Frozen Maze with Wormholes:

Task 1: Algorithm Implementation and Explanation

I implemented Value Iteration, a dynamic programming algorithm for solving Markov Decision Processes (MDPs). Unlike combinatorial search algorithms such as A*, Value Iteration is specifically designed for stochastic environments where action outcomes are probabilistic, making it ideal for this slippery frozen maze problem.

For the environment models, when user attempts action a with success probability p, the actual movement probability follows:

- Intended direction: p

- Perpendicular direction: 1-p

Movement mappings:

LEFT (0) → (0, -1)

DOWN (1) → (1,0)

RIGHT (2) → (0,1)

UP (3) → (-1,0)

Perpendicular mappings:

0→1

1→2

2→3

3→0

I modify the falling into holes does not terminate the episode, allowing continued navigation and enabling accurate hole-count statistics.

1. MDP Model Construction

I create a function build_transition_model() to pre-estimate all state transitions for efficiency. For each (state, action) pair, it calculates outcomes for both intended and perpendicular directions, handling boundary collisions, walls, and wormhole teleportation. Probabilities are merged when both directions lead to the same cell.

The reward structure guides optimal behavior:

- Goal: +100

- Hole: -5 (larger penalty)

- Normal step: -1 (common movement)

2. Value Iteration Algorithm

The algorithm iteratively updates state values using the Bellman optimality equation:

$$V^*(s) = \max_a \Sigma P(s'|s,a)[R(s,a,s') + \gamma V(s')]$$

For each state, it computes Q-values for all actions, selects the maximum, and updates both the value function V(s) and policy π(s). The discount factor $\gamma=0.99$ balances immediate and future rewards. Convergence occurs when the maximum value change (delta) drops below threshold $\theta=0.01$.

Experiment Results

Setup:

N=50

wall_prob=0.15

hole_prob=0.15

n_portals=15

p=0.9

Training: Converged in 620 iterations (delta: 6.61→0.012)

Execution:

92 steps

5 holes fallen

Performance P=460

Result:

Wormholes utilized effectively (steps 69, 73) and repeated holes fall near goal (steps 89-91 at position 49,48) due to stochasticity, and policy balanced hole avoidance with path efficiency.

## Task 2: Impact of Wormhole Density

Experimental Setup:

N=50

wall_prob=0.15

hole_prob=0.15

p=0.9

varying n_portals

| Number of Portals | Steps | Holes Fallen | Performance (P) |
|---|---|---|---|
| 5 | 133 | 1 | 133 |
| 10 | 85 | 2 | 170 |
| 15 | 87 | 2 | 174 |
| 20 | 52 | 1 | 52 |
| 25 | 61 | 2 | 122 |
| 30 | 55 | 4 | 220 |

Result:

Performance improves from 5 to 20 portals (133→52), achieving optimal efficiency at 20 portals. But its further increase to 30 portals degrades performance (P=220).

A moderate portal density (20 portals) provides beneficial shortcuts without incurring excessive risks. The optimal configuration balances the benefits of teleportation with the risk of exposure to dangerous areas. Beyond 20 portals, random placement increases the probability of teleporting to dangerous areas. This is particularly evident with 30 portals when the number of dangerous areas increases dramatically to four. Value iteration learns the utility of portals through value propagation: beneficial portals correspond to high-value exits, while harmful portals correspond to low-value exits.

Task 3: Impact of Hole Density

Experimental Setup:

N=50

wall_prob=0.15

n_portals=15

p=0.9

varying hole_prob

| Hole Probability | Step | Holes Fallen | Performance(P) |
|---|---|---|---|
| 0.05 | 61 | 1 | 61 |
| 0.10 | 79 | 2 | 158 |
| 0.15 | 94 | 3 | 282 |
| 0.20 | 65 | 11 | 715 |
| 0.25 | 32 | 5 | 160 |
| 0.30 | 62 | 8 | 496 |

Result:

Performance exhibits non-linear behavior. Dramatic spike at hole_prob=0.20 (P=715, 11 holes). Surprisingly, hole_prob=0.25 achieves better performance (P=160) with fewer steps (32).

This non-monotonic relationship reveals an adaptive strategy selection mechanism. At moderate densities (0.05 - 0.15), the algorithm can find safe paths. When the density reaches 0.20, the hole density reaches a critical threshold, making safe navigation extremely difficult and resulting in multiple encounters with holes along the way. In the range of 0.25 - 0.30, iteration of this value changes the strategy: strategic landings are chosen on "preferred holes" (closer to the target) to minimize the total path distance. This indicates that the algorithm makes a subtle trade-off between safety and efficiency. In some cases, a shorter but riskier path may be more efficient than a longer but safer path.

Task 4: Impact of Action Stochasticity

Experimental Setup:

N=50

wall_prob=0.15

hole_prob=0.15

n_portals=15

varying p

| p value | steps | Holes fallen | Performance(p) |
|---|---|---|---|
| 0.9 | 73 | 2 | 146 |
| 0.8 | 82 | 5 | 410 |
| 0.7 | 94 | 3 | 282 |
| 0.6 | 122 | 3 | 366 |
| 0.5 | 130 | 3 | 390 |
| 0.4 | 117 | 5 | 585 |
| 0.3 | 89 | 2 | 178 |
| 0.2 | 82 | 3 | 246 |

Result:

Performance degrades from p=0.9 (P=146) to p=0.4 (P=585), then improves at lower p values (0.3, 0.2). The worst performance occurs at p=0.4, handling a maximum of five holes.

High confidence (p=0.9) enables precise navigation and predictable results. As p values decrease to 0.4–0.5, the increased randomness leads to a challenging intermediate state: insufficient confidence for precise control and insufficient randomness to benefit from chance. The performance improvement at extremely low p values (0.2–0.3) suggests that very high randomness can produce beneficial random walks and potentially unintentional shortcuts. Value iteration adapts to increasing uncertainty by computing increasingly robust strategies and selecting paths with high probability of success. The robustness of the algorithm in the p∈[0.2, 0.9] range validates the MDP formula for random navigation.