AR525 Reinforcement Learning for Robotics

Assignment 1

Rishang Yadav (B23173)
Bhumika Gupta (B23036)

February 5, 2026

Analysis Report of Dynamic Programming Methods

This report presents a comparative analysis of **Policy Iteration (PI)** and **Value Iteration (VI)** applied to a 5×6 GridWorld navigation problem. The comparison focuses on convergence behavior, computational efficiency, path quality, and the effects of reward structure and stochastic transitions. All results are based on the experiments performed and plots generated in this assignment.

Comparison: Policy Iteration vs Value Iteration

Both Policy Iteration and Value Iteration aim to solve the same Bellman optimality equations and, as expected, converge to the **same optimal policy** in deterministic settings. However, their convergence dynamics and computational characteristics differ significantly.

Figure 1 shows the optimal policy obtained in the deterministic environment. Both methods produce identical directional policies that successfully avoid obstacles and guide the agent from the start (S) to the goal (G) along the shortest feasible path.

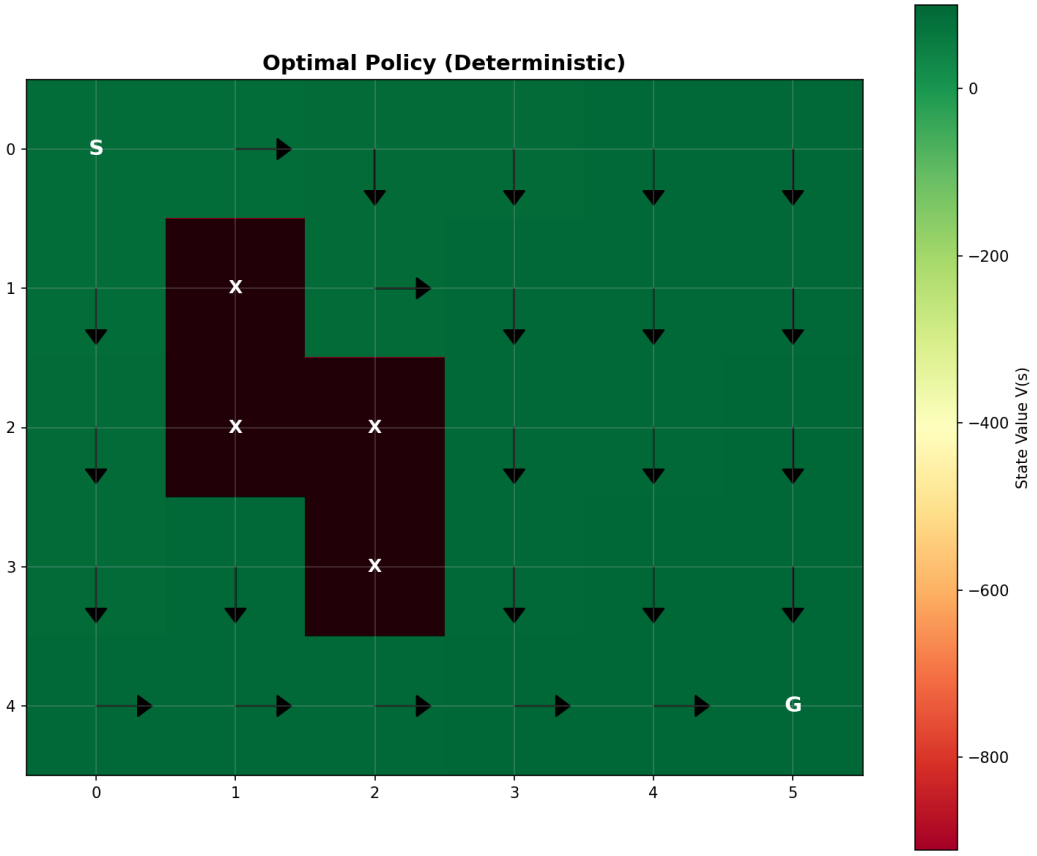

Figure 1: Optimal deterministic policy with obstacles, start (S) and goal (G).

Convergence Speed

Number of Iterations

Figure 2 compares the Bellman residuals of Policy Iteration and Value Iteration in a deterministic environment.

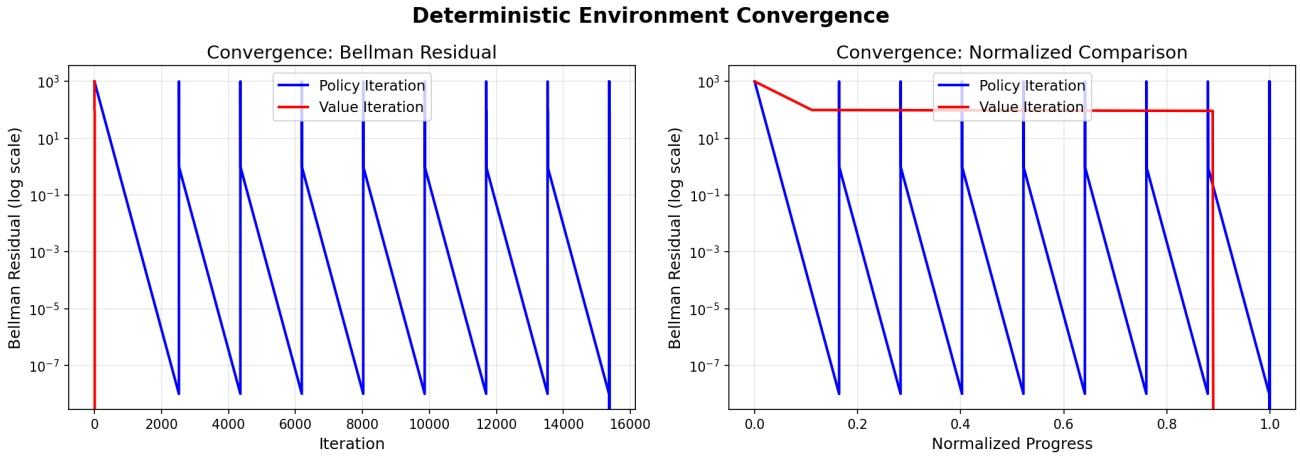

Figure 2: Convergence comparison of Policy Iteration and Value Iteration in a deterministic environment.

- **Policy Iteration** converges in a small number of policy updates (≈ 8–10).
- **Value Iteration** requires a larger number of iterations (≈ 24) to reach the same convergence tolerance.

The Bellman residual curve for Policy Iteration exhibits a characteristic **saw-tooth pattern**. This occurs because each policy evaluation step performs multiple Bellman backups, rapidly reducing the residual, followed by a reset when the policy is updated. In contrast, Value Iteration remains constant and then decreases sharply in the residual.

Insight: Policy Iteration converges in fewer outer iterations, while Value Iteration provides smoother and more stable convergence behavior.

Normalized Convergence Progress

The normalized comparison in Figure 2 highlights that Policy Iteration achieves most of its residual reduction early in the process, whereas Value Iteration requires sustained iterations throughout to converge.

Computation Time

Although exact timings depend on hardware, the following trends are consistently observed:

- **Policy Iteration** has a higher per-iteration cost due to repeated full policy evaluations.
- **Value Iteration** has a lower per-iteration cost, performing a single Bellman backup per state.

For the relatively small GridWorld used in this assignment, Policy Iteration is slightly more efficient overall due to its faster convergence in terms of iterations. However, Value Iteration is more predictable and often preferred in larger or more complex state spaces.

Path Quality

Both algorithms converge to the **same optimal path** in deterministic settings, as shown in Figure 1. The resulting path:

- Avoids all obstacle cells
- Has minimum path length
- Contains no loops or redundant moves

This confirms that differences between Policy Iteration and Value Iteration lie in **convergence behavior** rather than final solution quality.

Effect of Reward Structure

Figure 3 compares dense and sparse reward structures.

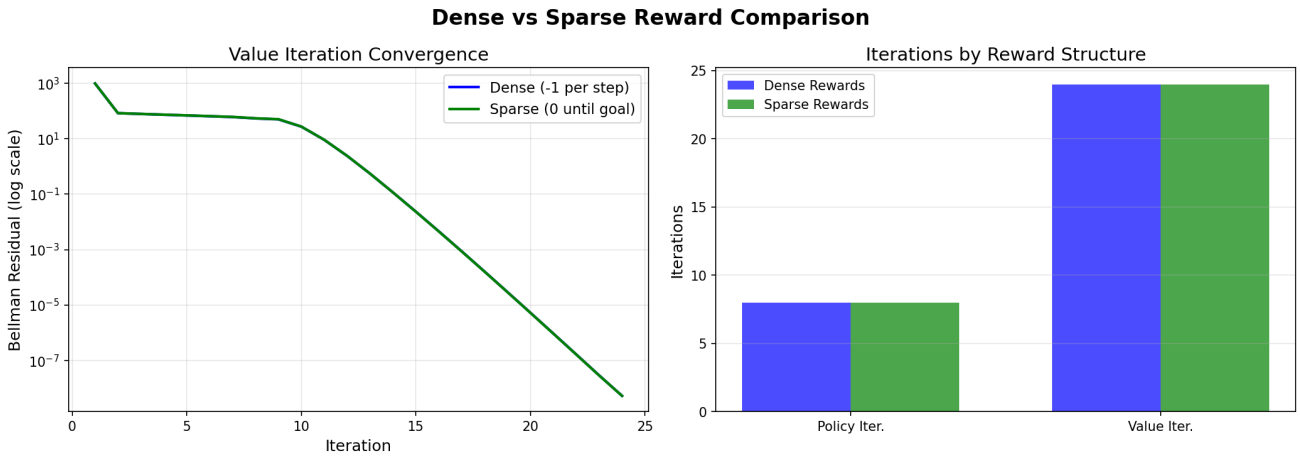

Figure 3: Effect of dense vs sparse rewards on convergence.

With dense rewards, the agent receives a negative reward at each step, providing continuous learning feedback, whereas in sparse reward settings, the agent receives reward only at the goal. As shown in Figure 3, dense and sparse rewards have nearly identical iteration counts (8 PI, 24 VI), but differ in convergence dynamics. This suggests that in well-structured environments, discount factor and convergence threshold dominate over reward shaping.

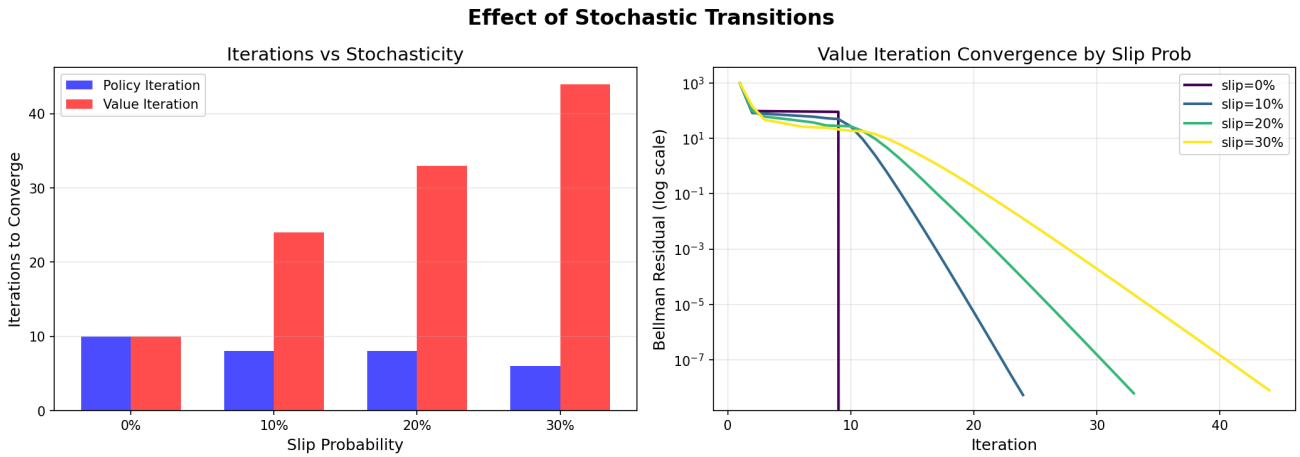

Figure 4: Effect of stochastic transitions (slip probability) on convergence behavior.

Effect of Stochastic Transitions

Figure 4 illustrates the effect of increasing slip probability on convergence.

- Value Iteration is more sensitive to transition uncertainty
- Policy Iteration remains relatively robust.

Higher slip probability introduces uncertainty in state transitions, Value Iteration shows 83% increase in iterations at 30% slip probability, while Policy Iteration remains robust. This highlights PI's strength in uncertain environments.

Summary of Observations

Aspect	**Policy Iteration**	**Value Iteration**
Iterations to converge	Fewer	More
Per-iteration cost	Higher	Lower
Residual behavior	Non-smooth	Smooth
Sensitivity to stochasticity	Lower	Higher
Sensitivity to sparse rewards	Lower	Higher
Final policy quality	Optimal	Optimal

Conclusion

Both Policy Iteration and Value Iteration successfully solve the GridWorld problem and converge to the same optimal policy. Policy Iteration is more efficient for small, fully-known environments, while Value Iteration offers stability and simplicity that scale better to larger problems. The experiments also demonstrate how stochastic transitions significantly influence convergence dynamics, whereas the reward structure does not seem to affect the algorithms.