A key idea in artificial intelligence is reinforcement learning (RL), which allows agents to interact with their surroundings to learn the best behaviors. The efficiency, convergence, and policy quality of RL algorithms can be directly compared in deterministic settings, where each action's result is predictable. In a grid-based setting with terminal rewards, this paper investigates three traditional algorithms: Value Iteration, Policy Iteration, and Q-Learning. The emphasis is on how each algorithm converges and how the learned policies and value functions are affected by the discount factor ($\gamma = 0.9, 0.5, 0.1$).

Value iteration is a dynamic programming technique that uses the Bellman optimality equation repeatedly to update state values. After converging in twelve iterations with a high discount factor ($\gamma = 0.9$), the algorithm generated policies that prioritized long-term planning and consistently sought the +20 reward state while avoiding penalties. The algorithm produced more cautious policies while converging in the same number of iterations by balancing immediate and future rewards at $\gamma = 0.5$. With $\gamma = 0.1$, the algorithm became shortsighted, focusing almost entirely on immediate outcomes and ignoring distant rewards, even though convergence still required twelve iterations.

Policy iteration alternates between assessing and refining a policy. It converged faster than Value Iteration when starting from a random policy. Convergence happened in five iterations at $\gamma = 0.9$, producing policies with strong long-term planning that were almost exactly the same as those from Value Iteration. Convergence also required five iterations at $\gamma = 0.5$, resulting in moderately cautious policies. Convergence was fastest at $\gamma = 0.1$, requiring only four iterations; however, the resulting policies were reactive and shortsighted, putting short-term gains ahead of long-term benefits.

In contrast to the other two, Q-Learning is a model-free algorithm that uses exploration to learn state-action values. It consistently converged in eight iterations across all discount factors in this deterministic setting. Long-term planning was reflected in Q-values at $\gamma = 0.9$, resulting in policies that were comparable to those from Value and Policy Iteration. The agent balanced short-term and long-term rewards at $\gamma = 0.5$. The agent rapidly converged to short-sighted policies at $\gamma = 0.1$, prioritizing immediate rewards over far-off high-value states. Q-Learning remained stable and effective even though it was marginally less efficient than Policy Iteration.

The convergence speed and discount factor sensitivity of the three algorithms differ noticeably. Value Iteration was the slowest, while Policy Iteration and Q-Learning were the fastest. Larger value magnitudes and long-term planning-focused policies were produced by higher discount factors. Reactive, short-term policies resulted from compressed values caused by lower discount factors. Despite these variations, when γ was high, all three algorithms generated consistent optimal policies; however, as γ decreased, they diverged.

To sum up, each of Value Iteration, Policy Iteration, and Q-Learning exhibits distinct advantages in deterministic settings. Q-Learning is adaptable and model-free, Policy Iteration is effective and reliable, and Value Iteration is principled but slow. Agent behavior is significantly influenced by the discount factor γ, which establishes whether policies give priority to immediate results or long-term rewards. Designing intelligent agents that fit particular planning horizons and reward structures requires an understanding of these dynamics.