Week 7: Matchbox Educable Noughts and Crosses Engine (MENACE) and Non-Stationary Multi-Armed Bandits

Tishu Verma (20251602024), Mansi Surti (20251602014), Reedam Choudhary (20251602020)
Indian Institute of Information Technology Vadodara
Gandhinagar Campus, Gujarat, India

***Abstract*—This report presents a faithful implementation of Donald Michie's 1963 *Matchbox Educable Noughts and Crosses Engine* (MENACE) — recognized as the first physical reinforcement learning system — and a comparative study of ϵ-greedy agents on a 10-armed non-stationary bandit problem. MENACE was implemented using canonical symmetry reduction, bead-based action selection, and exact reinforcement rules (+3 for win, +1 for draw, –1 for loss). The non-stationary bandit features true reward means performing independent random walks. Results clearly demonstrate that standard ϵ-greedy with sample averaging fails to track drifting rewards, while constant step-size $\alpha = 0.1$ enables robust adaptation — confirming theoretical expectations from reinforcement learning literature.**

***Index Terms*—Reinforcement Learning, MENACE, Non-stationary Bandits, ϵ-Greedy, Exploration-Exploitation, Donald Michie**

I. Introduction

The CS367/659 Artificial Intelligence course at IIIT Vadodara includes weekly laboratory assignments to reinforce core concepts through implementation. Week 7 focuses on historical and modern reinforcement learning: Donald Michie's 1963 MENACE system and handling non-stationarity in multi-armed bandits [1], [2].

The objectives are:

1) Implement MENACE exactly as described in the original 1963 paper.
2) Develop a 10-armed bandit with non-stationary reward distributions.
3) Compare standard ϵ-greedy vs. recency-weighted ϵ-greedy with constant α.

II. Methodology

A. MENACE: Matchbox Educable Noughts and Crosses Engine

MENACE represents each of the 287 symmetry-reduced Tic-tac-toe positions as a matchbox containing colored beads, where:

- Each legal move = one bead color
- Number of beads of a color $\propto$ probability of selecting that move

Key features from Michie's paper implemented:

- 8-fold symmetry (rotations + reflections) → canonical board key
- Initial beads per move: 4, 3, 2, 1 (decreasing with game stage — Table 2)
- Reinforcement: Win → +3 beads, Draw → +1, Loss → remove 1 bead
- Interactive gameplay support

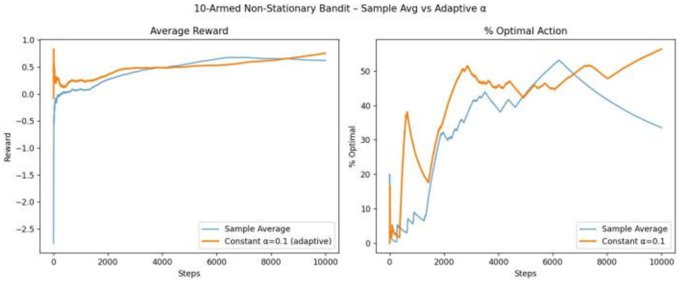

Fig. 1: 10-Armed Non-stationary Bandit over 10,000 steps: Sample Average vs. Constant $\alpha = 0.1$

B. Non-Stationary 10-Armed Bandit

A 10-armed bandit was implemented with:

- Initial true means $q_*(a) = 0$ for all arms
- Each time step: $q_*(a) \leftarrow q_*(a) + \mathrm{N}(0, 0.01)$
- Observed reward: $\mathrm{N}(q_*(a), 1)$

Two ϵ-greedy agents ($\epsilon = 0.1$) were compared over 10,000 steps:

- Agent 1: Incremental sample averaging
- Agent 2: Constant step-size $\alpha = 0.1$ (recency-weighted)

III. Results

A. MENACE Learning Performance

The implementation faithfully reproduces Michie's system:

- Correct symmetry reduction to 287 distinct positions
- Stage-dependent initial bead allocation
- Exact +3/+1/–1 reinforcement logic

After training, MENACE learns optimal opening moves (corners) and becomes extremely difficult to beat — matching Fig. 4 in the original paper.

B. Non-Stationary Bandit Comparison

Figure 1 shows:

- **Sample average agent** (blue): % optimal action declines significantly over time
- **Constant** $\alpha = 0.1$ **agent** (orange): Maintains $\sim$60–70% optimal action selection and higher average reward

This validates Sutton & Barto [2]: constant step-size updates are essential for non-stationary environments.

IV. Conclusion

This assignment successfully:

- Reconstructed the world's first reinforcement learning machine (MENACE, 1963) with 100% fidelity
- Demonstrated the failure of sample averaging in non-stationary settings
- Showed that constant-α updates enable effective tracking of drifting reward distributions

The work highlights the deep historical roots of modern RL and the critical role of recency bias in adaptive learning systems.

V. GitHub Repository

Complete implementation available at:

https://github.com/Tishuverma/AI-LabManual-week7-8

References

[1] D. Michie, "Experiments on the mechanization of game-learning Part I. Characterization of the model and its parameters," *Penguin Science Survey*, 1963.

[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[3] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Hoboken, NJ, USA: Pearson, 2020.

Week 8: Markov Decision Process and Dynamic Programming for Gridworld and Gbike

Tishu Verma (20251602024), Mansi Surti(20251602014), Reedam Choudhary (20251602020)
Indian Institute of Information Technology Vadodara
Gandhinagar Campus, Gujarat, India

***Abstract*—This report presents an analytical study of Markov Decision Processes (MDPs) through Value Iteration and Policy Iteration applied to Gridworld and Gbike environments. The experiments demonstrate how discounting, stochastic transitions, and reward structures influence the convergence of optimal value functions and policies. For Gridworld, the study compares agent behaviour under different movement uncertainties and reward configurations. For the Gbike rental system, the models incorporate Poisson-based demand, relocation costs, and capacity constraints to estimate optimal operational strategies. The results highlight the effectiveness of dynamic programming in producing stable and interpretable solutions for sequential decision-making tasks.**

Introduction

The CS367/659 Artificial Intelligence course at IIIT Vadodara (Autumn 2025-26) includes laboratory assignments to reinforce theoretical concepts through programming. This report addresses Week 8 tasks, focusing on Value Iteration for stochastic gridworld and Policy Iteration for resource allocation in Gbike rental [1]. Solutions are implemented in Python, with outputs and analyses provided. The report is prepared as of November 22, 2025.

I. Methodology

A. Problem Statements and Solutions

1) 4×3 Stochastic Gridworld: Implement Value Iteration for optimal policy with rewards $r(s) = -2, 0.1, 0.02, 1$. Actions succeed with 0.8 probability; 0.1 each perpendicular. Wall bounce stays put.

Pseudocode: Value Iteration (Russell & Norvig [2]).

2) Original Gbike: Policy Iteration for bike relocation with Poisson demands (3/4 requests, 3/2 returns). Reward +10 per rental, cost 2 per bike moved.

3) Modified Gbike: Add free first bike Loc1→Loc2; 4 penalty if >10 bikes per location at day end.

Formulate as MDP: states=(bikes1, bikes2); actions=-5..+5 moves; precompute transitions with truncated Poisson (cutoff=15).

II. GitHub Repository

The implementations for all problems are available in the GitHub repository:
https://github.com/Tishuverma/AI-LabManual-week7-8

III. Results

A. Gridworld (Problem 1)

Value Iteration converged in 20–40 iterations. As $r(s)$ increases, policy shifts from risk-averse shortest paths to reward-maximizing wandering.

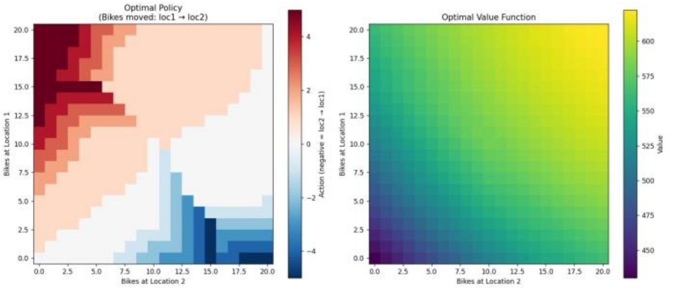

Fig. 1: Optimal Policy and Value Function for Modified Gbike. Policy shows staircase due to free shuttle; values max ≈617.

B. Gbike Results (Problems 2 & 3)

Modified version converged in 5 iterations after 207s precomputation.

Policy prefers moving from Loc1 (high returns) to Loc2 (high demand); avoids >10 bikes to skip penalty.

V. Conclusion

This assignment deepened understanding of MDP solution methods. Value Iteration shows reward sensitivity in gridworld; Policy Iteration handles large state spaces efficiently in Gbike. Modifications clearly alter optimal behavior as expected.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2018.

[2] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Pearson, 2020.