MDP Applications: Grid World and Gbike

Divyansh Nahar
Computer Science and Engineering
202351035@iiitvadodara.ac.in

Jinendra Kumar Jain
Computer Science and Engineering
202351050@iiitvadodara.ac.in

Kuldeep Purbiya
Computer Science and Engineering
202351072@iiitvadodara.ac.in

***Abstract*—This report investigates the application of Value Iteration to solve a Markov Decision Process (MDP) for a 4x3 stochastic grid environment. The agent selects actions to maximize rewards while navigating towards terminal states with rewards of +1 and -1, accounting for probabilistic outcomes and penalties for non-terminal moves. By iteratively computing the value function, the study determines the optimal policy, highlighting the effectiveness of dynamic programming techniques in decision-making under uncertainty.**

***Index Terms*—Hopfield Network, Associative Memory, Error Correction, Combinatorial Optimization, Eight-Rook Problem, Traveling Salesman Problem, Neural Network Dynamics, Energy Minimization**

I. Introduction

Markov Decision Processes (MDPs) provide a robust mathematical framework for solving decision-making problems in uncertain environments, making them ideal for resource management scenarios. This report addresses two distinct problems modeled as MDPs: navigating a stochastic grid world and managing resources in a bicycle rental business. In the first problem, an agent navigates a 4x3 grid envi- ronment to reach terminal states with rewards of +1 or -1. The agent selects actions (Up, Down, Left, Right) with an 80random orthogonal movements. Non-terminal moves incur a penalty of -0.04. Using Value Iteration, the value function and optimal policy are determined to maximize cumulative rewards, showcasing how dynamic programming effectively handles decision-making under uncertainty. The second problem involves managing bicycle rentals at two Gbike locations. Customer rental demands and returns at each location are modeled as Poisson random variables. The goal is to maximize daily revenue while minimizing costs associated with transferring bicycles between locations. Constraints include a maximum of 20 bicycles per location and a transfer limit of 5 bikes per night. Additional com- plexities include a free bike transfer by an employee and parking costs of INR 4 when more than 10 bikes are stored overnight at a location. Policy Iteration is applied to iteratively refine decisions, balancing stochastic demands, transfer costs, and parking constraints to optimize resource allocation and profitability. Both problems highlight the versatility of MDPs and dy- namic programming techniques in solving real-world decision- making challenges.

II. Objectives

The primary objective of this report is to apply Markov Decision Processes (MDPs) and dynamic programming techniques to solve decision-making problems in stochastic environments. The report aims to achieve the following:

1. Formulate the Problem as an MDP

Define the problems in terms of states, actions, rewards, and transition probabilities:

- For the 4×3 grid navigation problem, model the environment as a stochastic MDP where the agent's actions have probabilistic outcomes and rewards depend on the destination state.
- For the Gbike bicycle rental problem, establish an MDP capturing rental dynamics, transfers, parking limits, transfer costs, and stochastic demand/returns using Poisson distributions.

2. Implement Dynamic Programming Techniques

Solve the formulated MDPs using dynamic programming:

- Apply Value Iteration to compute the value function and optimal policy for the grid navigation problem, iterating until convergence and maximizing cumulative rewards.
- Use Policy Iteration for the Gbike problem, including policy evaluation and policy improvement steps, to optimize bicycle allocation and minimize costs.

3. Integrate Real-World Constraints

Incorporate practical complexities into both MDP models:

- Introduce penalties for non-terminal movements in the grid navigation problem to simulate real-world inefficiencies.
- For the Gbike problem, integrate free transfers by employees, additional parking costs for exceeding limits, and transfer capacity restrictions to reflect operational challenges.

4. Analyze and Compare Results

To evaluate the effectiveness of the approaches and derive insights:

- Assess the convergence behavior of Value Iteration and interpret the resulting value function and optimal policy for the grid navigation problem.
- Compare the initial and optimal policies derived through Policy Iteration for the Gbike problem, analyzing the impact of various constraints on profitability and operational efficiency.

III. Theoretical Framework

This section provides the theoretical foundation for solving the given problems using Markov Decision Processes (MDPs), including their formulation, solution techniques, and specific considerations for both the Grid World and Gbike rental problems.

A. Markov Decision Processes (MDPs)

An MDP is formally defined by the tuple (S, A, P, R, γ), where:

- S: Set of states representing all possible configurations of the system.
- A: Set of actions available to the agent in each state.
- $P(s' \mid s, a)$: Transition probability of moving to state s' from state s after taking action a.
- $R(s, a)$: Immediate reward received after transitioning from s to s' using action a.
- γ: Discount factor $(0 \leq \gamma \leq 1)$ that prioritizes immediate rewards over future rewards.

The objective is to compute an optimal policy $\pi(s)$ that maximizes the cumulative discounted reward.

B. Dynamic Programming for Solving MDPs

Dynamic programming techniques such as Value Iteration and Policy Iteration are employed:

- **Value Iteration**: Iteratively updates the value function $V(s)$ using the Bellman Optimality Equation:

$$V(s) = \max_a \left[R(s, a) + \gamma \sum_{s'} P(s' \mid s, a) V(s') \right]$$

- **Policy Iteration**: Alternates between:
  - *Policy Evaluation*: Computing $V^\pi(s)$ for a given policy.
  - *Policy Improvement*: Updating the policy to maximize expected return.

  This process repeats until convergence to the optimal policy.

C. Stochastic Grid World Problem

The agent navigates a 4×3 grid environment with stochastic action outcomes. The value function $V(s)$ is computed using Value Iteration for different movement penalties $r(s)$.

- Each action succeeds with 80% probability.
- With 20% probability, the agent moves in a perpendicular direction.
- Non-terminal states incur a movement penalty $r(s)$.
- Terminal states yield rewards of $+1$ and -1.

D. Gbike Bicycle Rental Problem

The Gbike problem models the rental dynamics of bicycles at two locations as a continuing MDP.

- **States**: (s_1, s_2) where s_1 and s_2 denote the number of bikes at the two locations.
- **Actions**: Net number of bikes moved overnight between locations.
- **Rewards**:
  - INR 10 per rented bike,
  - INR 2 per bike moved (beyond the free one),
  - Parking penalties for exceeding 10 bikes.
- **Transitions**: Determined by Poisson-distributed rental requests and returns.

Policy Iteration is used to compute the optimal policy while accounting for constraints such as free bike transfers, parking limits, and movement costs.

IV. Solutions

A. Stochastic Grid World Problem

Value functions $V(s)$ corresponding to different movement penalties $r(s)$ were computed using Value Iteration. The results are summarized as follows:

- $r(s) = -2$: The high penalty encourages the agent to navigate quickly toward terminal states.
- $r(s) = 0.1$: The low penalty promotes exploratory behavior.
- $r(s) = 0.02$: The agent exhibits a balanced strategy, trading off exploration and exploitation.
- $r(s) = 1$: Positive rewards for movement discourage reaching terminal states, shifting behavior away from goal-directed navigation.

B. Gbike Bicycle Rental Problem

Policy Iteration was implemented to solve the Gbike problem with the following considerations:

- One bicycle can be transferred for free each night by an employee.
- An additional parking cost of INR 4 is incurred when more than 10 bicycles are kept at any location overnight.

The optimal policy maximizes daily profit by effectively balancing rental revenues, transfer costs, and parking penalties. Constraints such as transfer limits and Poisson-based stochastic rental demands were successfully incorporated into the model.

Code Implementation

Python code was developed to solve both MDP problems:

- **Value Iteration** for the grid world problem, updating the value function $V(s)$ until convergence.
- **Policy Iteration** for the Gbike problem, repeatedly performing policy evaluation and policy improvement.

Note: GitHub link for the full implementation: CS307-AI GitHub Repository – LAB-8

V. Conclusion

This report demonstrated the application of Markov Decision Processes (MDPs) and dynamic programming techniques to solve sequential decision-making problems under uncertainty. The Stochastic Grid World problem illustrated how Value Iteration can compute optimal value functions for

different movement penalties, showing how reward structures influence agent behavior in a probabilistic environment.

The Gbike bicycle rental problem was formulated as a continuing finite MDP, with states representing the number of bicycles at each location at the end of each day, actions corresponding to overnight transfers, and transitions governed by Poisson-distributed rental requests and returns. Policy Iteration was implemented to evaluate and improve policies until convergence.

The required real-world constraints were successfully integrated into the model:

- one free nightly transfer from location 1 to location 2 by an employee,
- a movement limit of five bicycles per night,
- a parking capacity of twenty bicycles at each location,
- and an additional INR 4 parking penalty whenever more than ten bicycles are kept overnight at any location.

The resulting optimal policy balances rental revenue, transfer costs, the benefit of the free transfer, and storage penalties. Overall, the analysis highlights the effectiveness of MDP formulations and Policy Iteration in solving operational decision-making problems involving stochastic demand and constrained resources.

REFERENCES

1) S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed., Pearson, 2022.
2) R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., MIT Press, 2018.
3) M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, Wiley, 2005.
4) C. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3–4, pp. 279–292, 1992.
5) D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 4th ed., Athena Scientific, 2017.
6) A. Gosavi, *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*, Springer, 2015.
7) T. Dietterich, "Hierarchical Reinforcement Learning with the MAXQ Value Function Decomposition," *Journal of Artificial Intelligence Research*, vol. 13, pp. 227–303, 2000.
8) M. Ghallab, D. Nau, and P. Traverso, *Automated Planning: Theory and Practice*, Morgan Kaufmann, 2004.