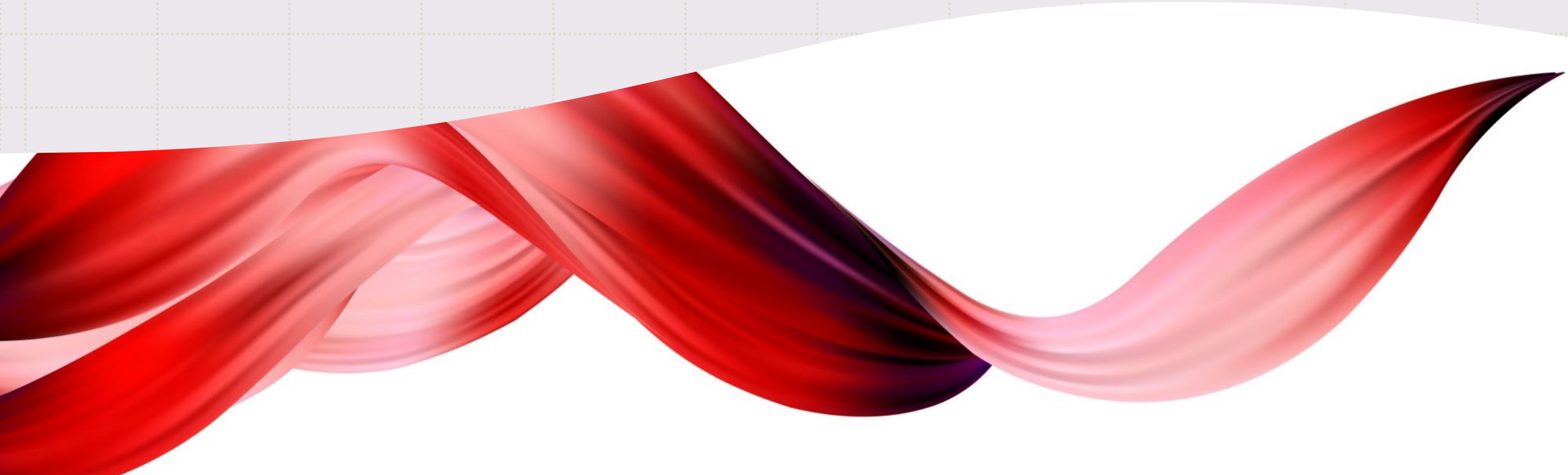
Wentworth Institute of Technology

COMP5700 – Classical AI

Markov Decision Process

Lev Sukherman



Markov Property

- **Memorylessness**
- Only the current state and the expected outcomes of possible actions taken from this state, without having to consider the path taken to reach the current state.

- $P(X_{t+n} = x \,|X_t, X_{t-1}X_{t-2}X_{t-k}) = P(X_{t+n} = x \,|X_t)$

Uncertainty

- **uncertainty** refers to situations in which the outcomes or conditions of an event or phenomenon cannot be precisely predicted or are not completely known

- Decision Making Under Uncertainty – whole area that includes lots of different techniques (Markov Decision Process, Bayesian Network, Markov Chain Monte Carlo (MCMC), N-step Bootstrapping, Q-Learning, Sarsa)

Markov Decision Process

- A Markov Decision Process is a framework used for modeling decision-making in situations where outcomes are partly random and partly under the control of a decision maker.

- States (S) – Possible configurations your system or environment can be in.
- Actions (A) - Various actions available to the decision-maker.
- Transition Probability (P) - Probability that a particular action taken in a current state.
- Rewards (R) – Action -> new state, receive the reward.
- Discounted Factor (γ) - A factor between 0 and 1 used to weigh the importance of future rewards .

Value Iteration

- $V_{i+1}(S) = R(S) + \gamma \underset{a \in A}{max} \sum_{s'}^{S} \{T(S, a, S') * V_i(S')\}$

- R(S) – the reward of achieving a state.
- T(S,a,S') – transition table (probability of transition to the state).
- V(S') – value of the successor state.

- Repeatedly updating the value of each state in the MDP using the Bellman Optimality Equation until the values converge to a stable set.

*Repeat till converged

Policy iteration

- $V^{\pi k}(S) = R(S) + \gamma \sum_{S'}^{S} \{T(S, \pi_k(S), S') * V^{\pi k}(S')\}$ (Policy evaluation)
- $\pi(S) = \underset{a \, \epsilon \, A}{argmax} \sum_{S'}^{S} \{T(S, a, S') * V^{\pi}(S')\}$ (Policy improvement)
- Policy Check: Determine whether the policy has changed.

- T(S, $S(\pi)$,S′) – transition table (probability of transition to the state using exact action).
- V^{π}(S′) – value using exact action.

Real-Time Dynamic Programing

- Computing or improving the policy online, as the agent interacts with the environment, when pre-computation of the full policy is computationally expensive.

- $V = R(S) + \gamma \sum_{s'}^{S} \{T(S, S(\pi), S') * V(S')\}$

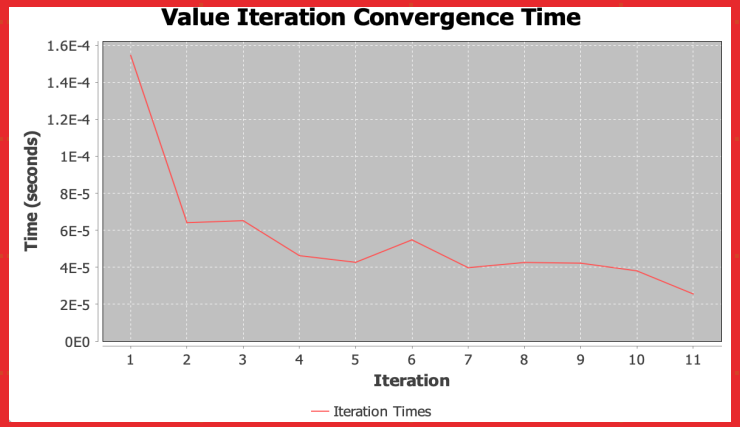
- Bellman update: Only the value functions for the visited states is updated using the Bellman equation.

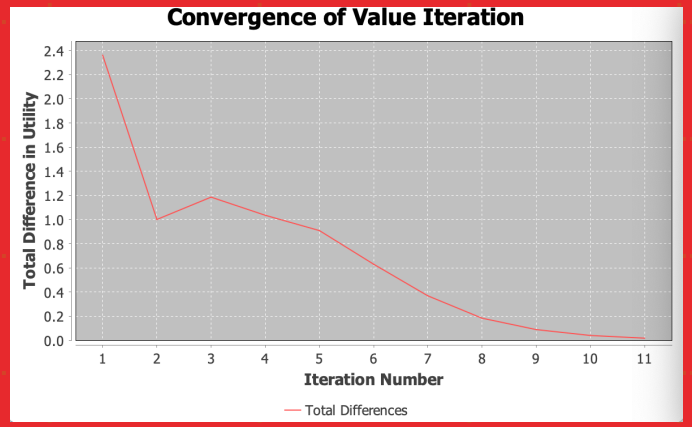
- Greedy Policy Improvement: After each update, the policy can be improved by choosing actions that maximize the current value function.

Results (Utility)

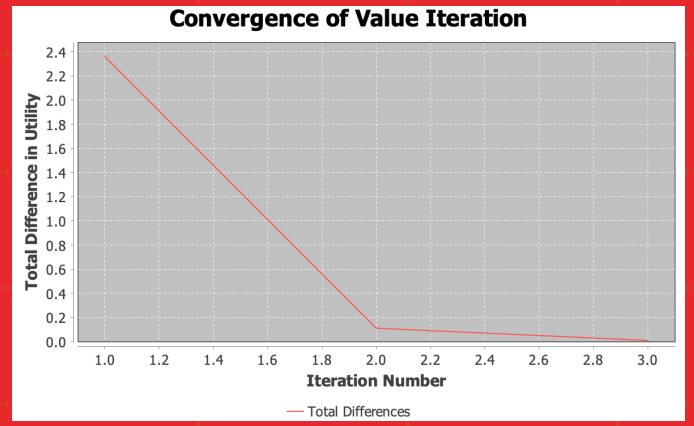
Initial state:
10 states
2 terminal states with reward of -1 and +1

$\gamma - discounted\ factor = 0.9$

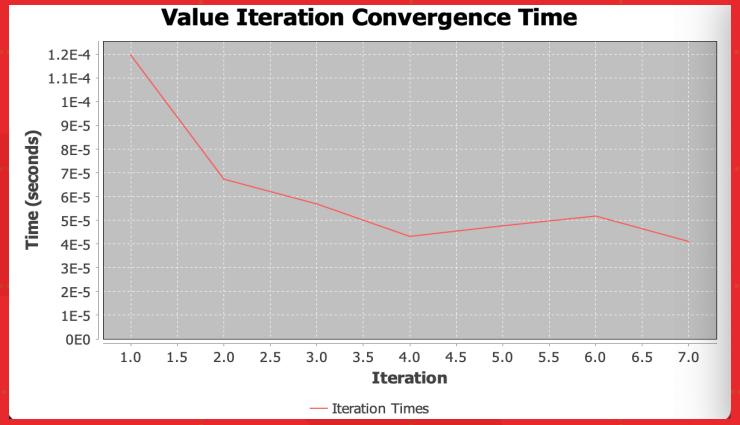


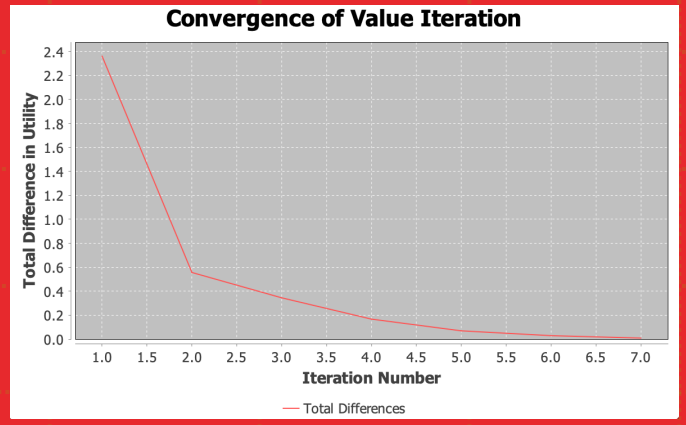


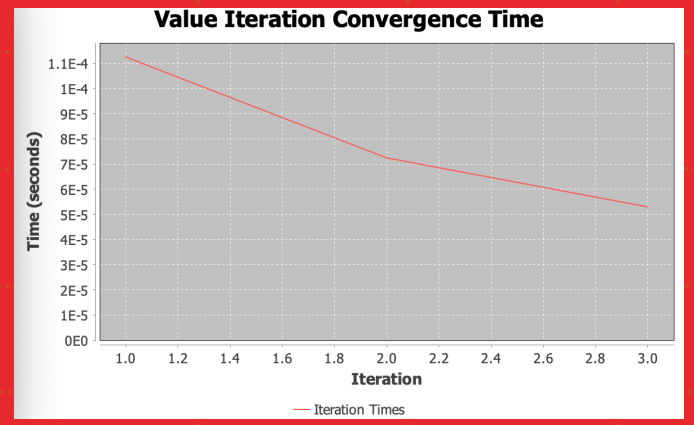
$\gamma - discounted\ factor = 0.1$



$\gamma - discounted\ factor = 0.5$



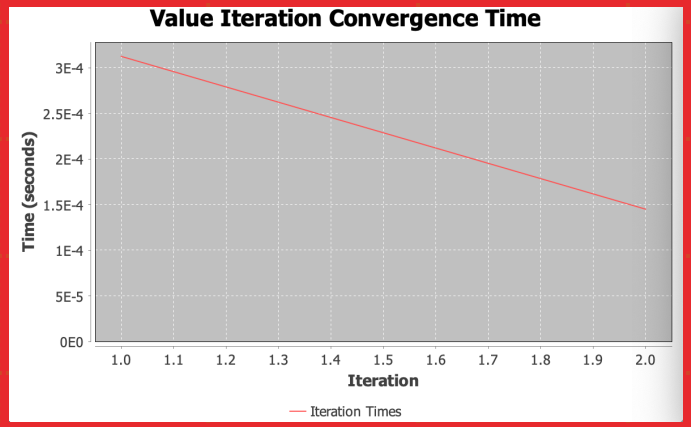


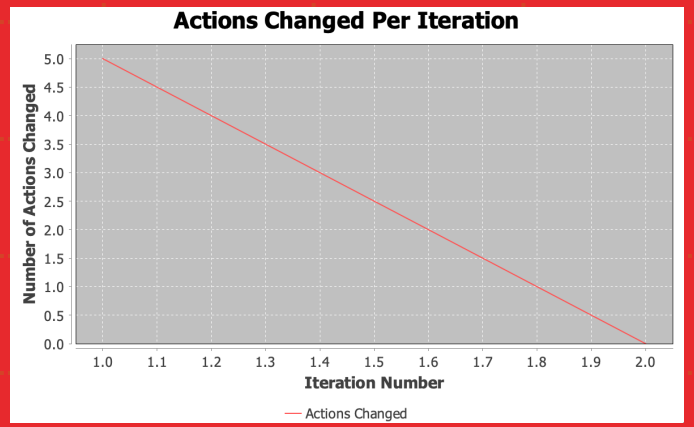


Results (Policy)

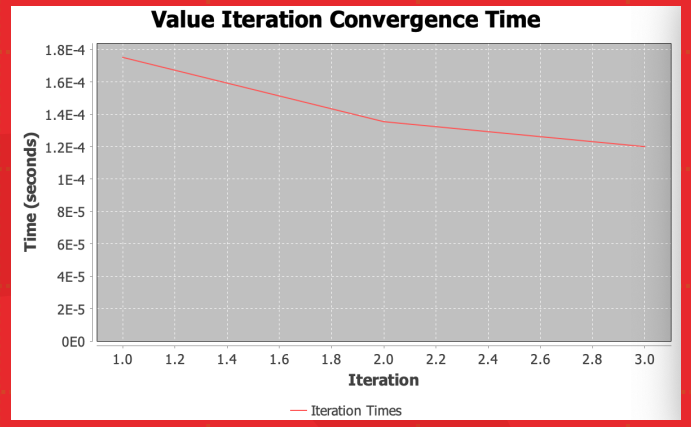
Initial state:
10 states
2 terminal states with reward of -1 and +1

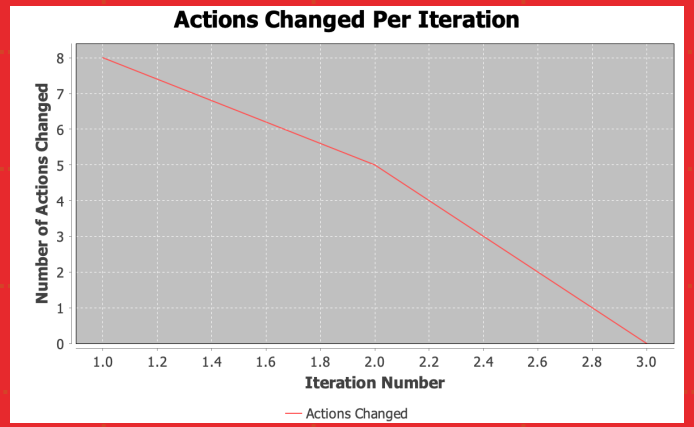
$\gamma - discounted\ factor = 0.9$



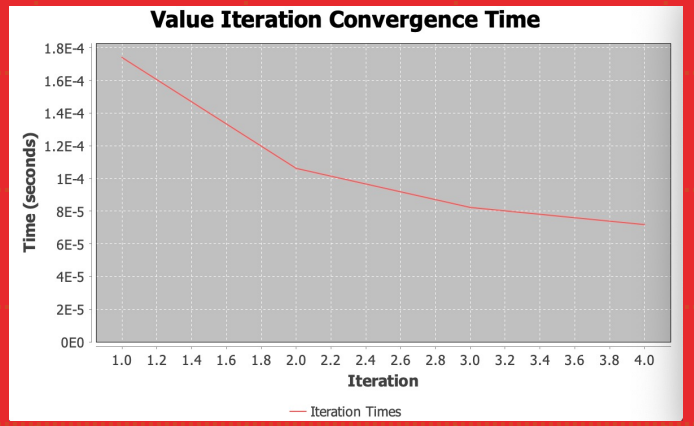


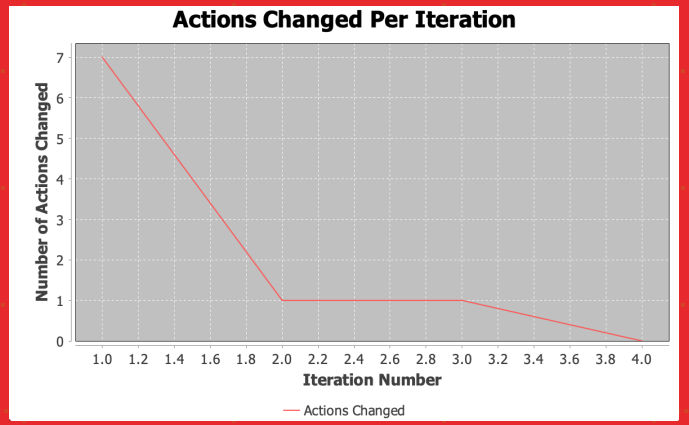
$\gamma - discounted\ factor = 0.5$





$\gamma - discounted\ factor = 0.1$





Results (Real-Time Dynamic Programing)

$\gamma - discounted\ factor = 0.9$

```
Iteration 0: Policy changes = 6
Iteration 1: Policy changes = 6
Iteration 2: Policy changes = 3
Iteration 3: Policy changes = 3
Iteration 4: Policy changes = 0

RTDP completed after 4 iterations.
Utility of state 0: -0.010108789397262796
Utility of state 1: 0.11910314335293865
Utility of state 2: 0.2894668332275463
Utility of state 3: -0.17451504216040004
Utility of state 4: -0.11953732899280001
Utility of state 5: 0.48217271780087406
Utility of state 6: -1.0
Utility of state 7: -0.11290155960935203
Utility of state 8: 0.61393019426816
Utility of state 9: 0.7921749719247873
Utility of state 10: 1.0
```

Initial state:
10 states
2 terminal states with reward of -1 and +1

$\gamma - discounted\ factor = 0.5$

```
Iteration 0: Policy changes = 6
Iteration 1: Policy changes = 6
Iteration 2: Policy changes = 2
Iteration 3: Policy changes = 3
Iteration 4: Policy changes = 2
Iteration 5: Policy changes = 0

RTDP completed after 5 iterations.
Utility of state 0: -0.0671828338761605
Utility of state 1: -0.07219697653125001
Utility of state 2: -0.027280655076562493
Utility of state 3: -0.07661612500000001
Utility of state 4: -0.04491656209826089
Utility of state 5: 0.06464800000000004
Utility of state 6: -1.0
Utility of state 7: 0.007208364699528548
Utility of state 8: 0.12547012068000002
Utility of state 9: 0.3823498947051251
Utility of state 10: 1.0
```

$\gamma - discounted\ factor = 0.1$

```
Iteration 0: Policy changes = 6
Iteration 1: Policy changes = 15
Iteration 2: Policy changes = 0

RTDP completed after 2 iterations.
Utility of state 0: -0.044442030513721954
Utility of state 1: -0.04441538209532149
Utility of state 2: -0.04437088416601331
Utility of state 3: -0.044436324
Utility of state 4: -0.04444068548607544
Utility of state 5: -0.043526508905601544
Utility of state 6: -1.0
Utility of state 7: -0.04440494496988866
Utility of state 8: -0.044000000000000004
Utility of state 9: 0.039964377930464624
Utility of state 10: 1.0
```

Conclusion

Complexity:

- Value Iteration:
  - Complexity/iteration: $O(|S^2||A|)$
  - Convergence: Linear
- Policy Iteration:
  - Complexity/iteration: $O(|S^3||A| + |S^2||A|)$
  - Convergence: Linear-Quadratic
- Real-Time Dynamic Programing:
  - Complexity/iteration: $O(|S||A|)$
  - Convergence: Stochastic Approximation

Artificial Intelligence and Machine Learning

- MDPs underpin many methods in reinforcement learning, where agents learn optimal policies for decision-making by interacting with an environment.

Examples:

- Video game AI, where NPCs (non-player characters) decide on actions based on the current game state.
- Autonomous vehicles that must make continuous driving decisions in an environment with other drivers and pedestrians.

Thanks for attention.

Any questions?