

# Advanced Topics in Reinforcement Learning

Lecture 18: Hierarchical RL I

Josiah Hanna

University of Wisconsin — Madison

# Announcements

- Midterm exam on Thursday (12/6) IN CLASS — be on time!
- Read Chapter 1 and 5 of “Multi-agent RL: Foundations and Modern Approaches”
- Mid-course Survey
- Phil Thomas Talk Wednesday: “Qualia Optimization: Exploring Mathematical Formulations of AI Experience”

# Learning Outcomes

After today, you will be able to:

1. Formulate the options framework to represent hierarchical RL problems.
2. Explain each piece of the options framework.
3. Translate value functions and update rules into semi-MDPs.

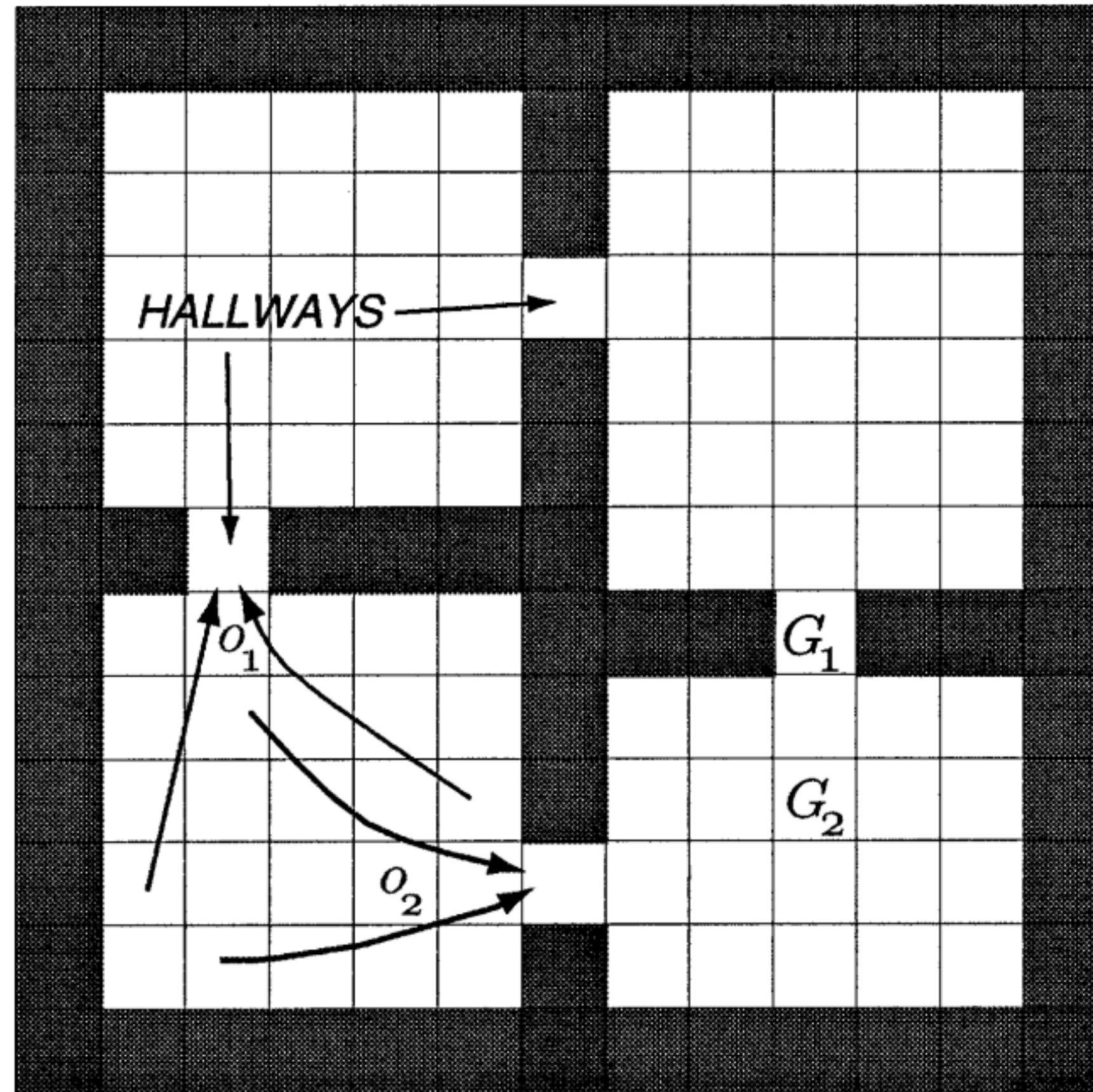
# Motivation for Abstraction

- People complete tasks by planning, learning, and acting at different level of abstraction.
  - Aids credit assignment and exploration.
- Behaviors are modular and re-used across tasks.
  - Transfer learning; subtask learning
- Different states may be functionally the same.

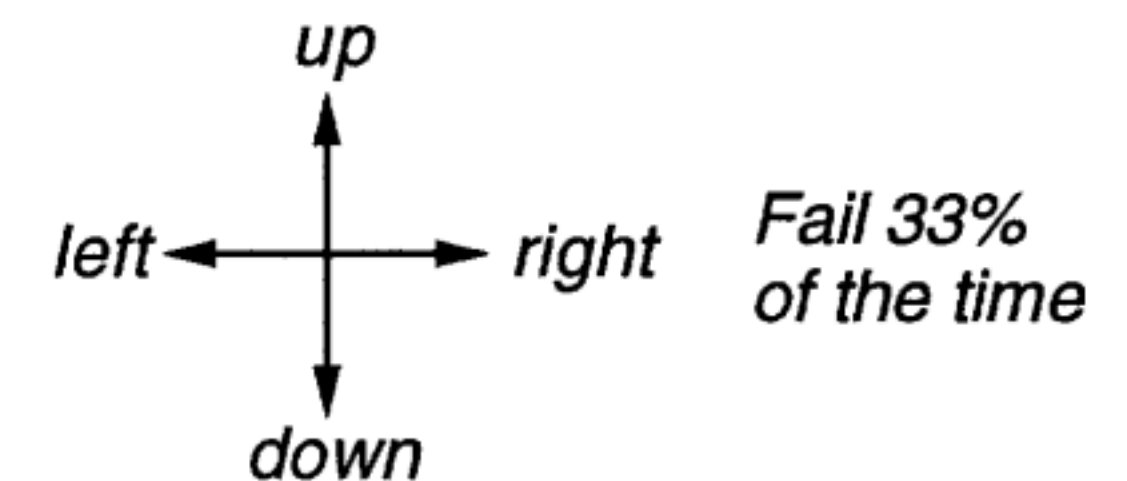


# Types of Abstraction

- State Abstraction
- Temporal Abstraction



*4 stochastic primitive actions*



*8 multi-step options  
(to each room's 2 hallways)*

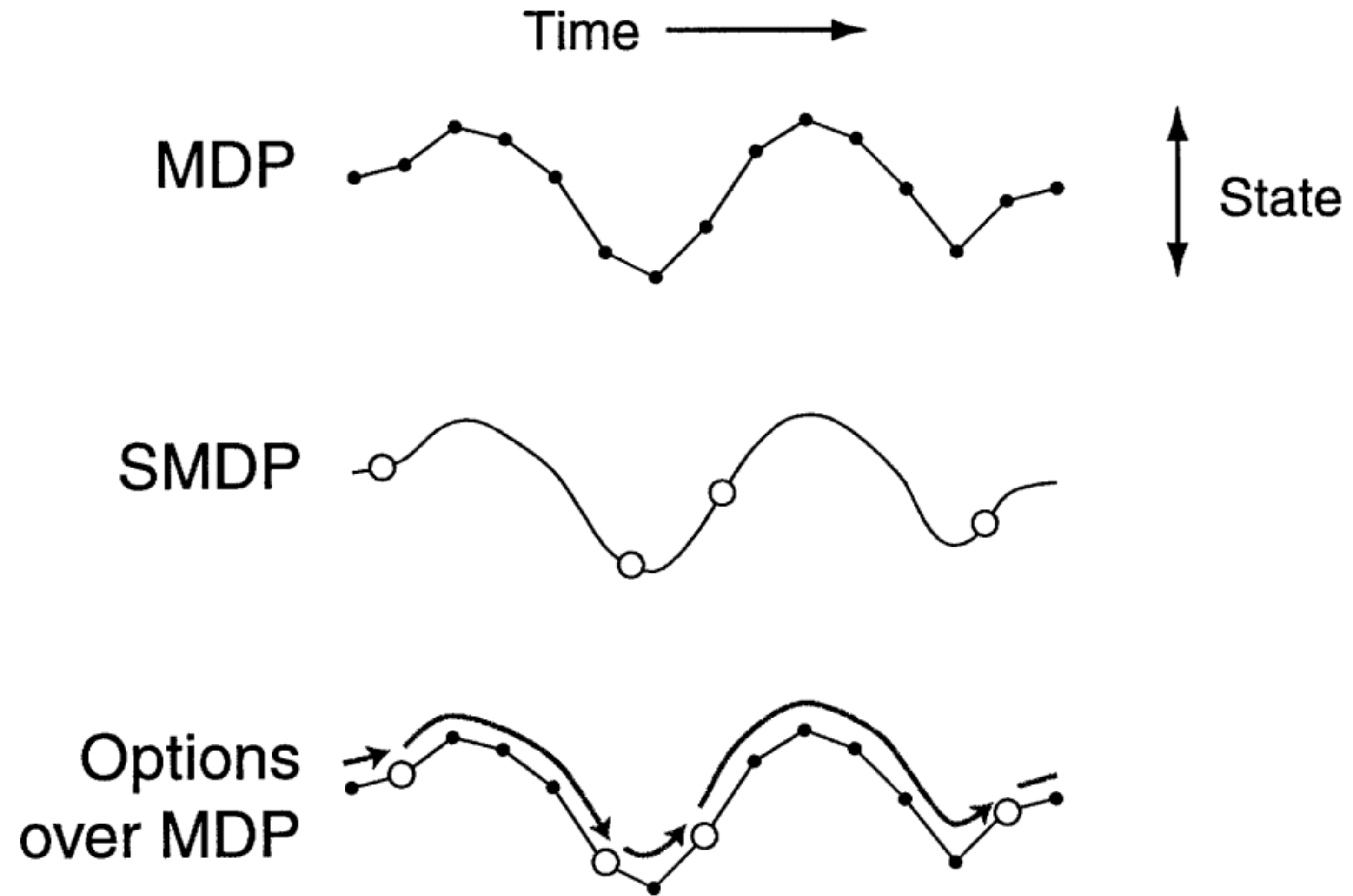
# Semi-MDPs

- Generalization of MDPs that allow actions to last for varying amounts of time.
  - $\mathcal{S}$ : same state set.
  - $\mathcal{A}$ : Action set.
  - $p(s', r, t | s, a)$ : Probability of transitioning to  $s'$  in  $t$  steps and collecting reward  $r$  along the way.
  - $\gamma$ : discount factor.
- $$v_{\pi}(s) = \sum_a \pi(a | s) \sum_{s', r, t} p(s', r, t | s, a) [r + \gamma^t v_{\pi}(s')]$$

# Options Framework

- Minimal change to MDP formalism to permit temporal abstraction.
- An option is:
  - $\Omega$ : Set of all possible histories.
  - $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0,1]$ : a Markov policy or  $\pi : \Omega \times \mathcal{A} \rightarrow [0,1]$ : a semi-Markov policy.
  - $\beta : \mathcal{S} \rightarrow [0,1]$ : termination probability.
  - $I \subset \mathcal{S}$ : initiation set.
- If the agent is in  $s \in I$ , then it can execute option  $o$ , which means following  $\pi$  until termination with probability  $\beta(S_t)$  at each step  $t$ .
- $\mathcal{E}(\pi, s, t)$ : the event that option  $\pi$  was initiated at time  $t$  in state  $s$ .

# Options Framework



# Policies

- Markov policies over options:  $\mu : \mathcal{S} \times \mathcal{O} \rightarrow [0,1]$ .
- Policies over options determine an underlying flat policy which outputs primitive actions.
- Flat policy is (usually) non-Markovian. Why?
  - Action selection at each time-step depends on what option is being executed and thus history.
  - Not fully non-stationary policies; only depend on history since current option was selected.

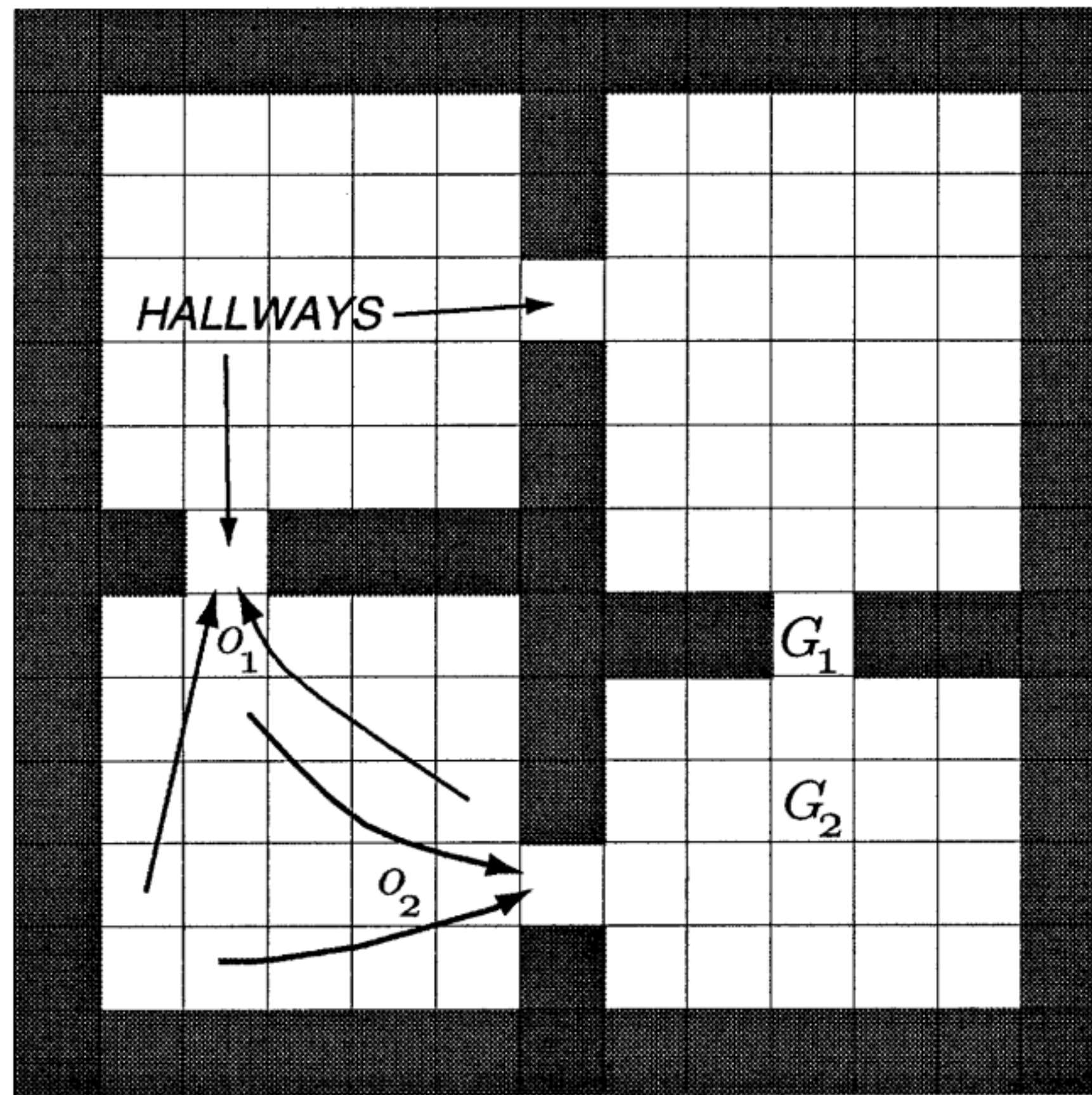
# Multi-Time Models and Value Functions

- $r_s^o = \mathbf{E}[R_{t+1} + \gamma R_{t+2} + \dots + \gamma^k R_{t+k} \mid \mathcal{E}(o, s, t)]$   
 $k$ : time where  $o$  terminates
- $p_{ss'}^o = \sum_{k=1}^{\infty} p(s', k) \gamma^k$  : Probability of transitioning in any number of time-steps.
- Action-value functions become *option-value* functions:
  - $q_{\mu}(s, o) = \mathbf{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \mid \mathcal{E}(o\mu, s, t)]$ .
  - $q_{\mu}(s, o) = r_s^o + \sum_{s'} p_{ss'}^o \sum_{o' \in \mathcal{O}} \mu(s', o') q_{\mu}(s', o')$

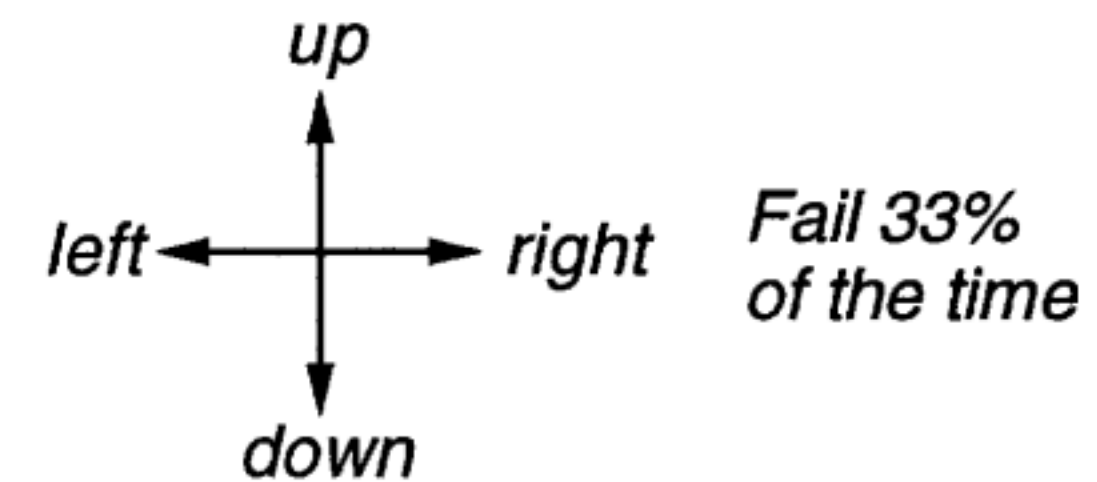
# Learning and Planning

- With recursive action-value definitions, learning and planning easily extend from MDPs:
  - Use value iteration with a known or learned  $p_{ss'}^o$  and  $r_s^o$ .
  - Use Q-learning for model-free learning.

# Examples



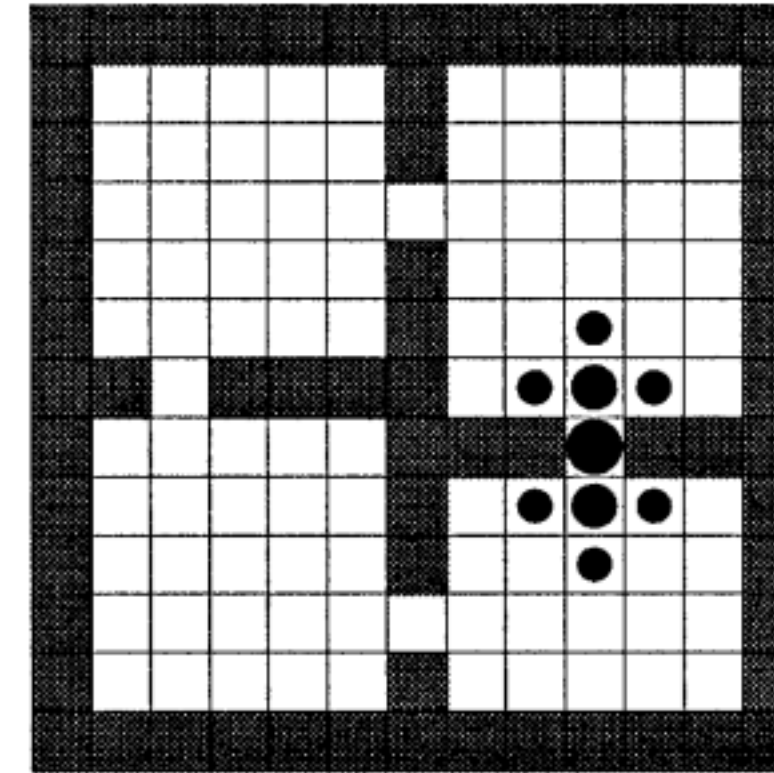
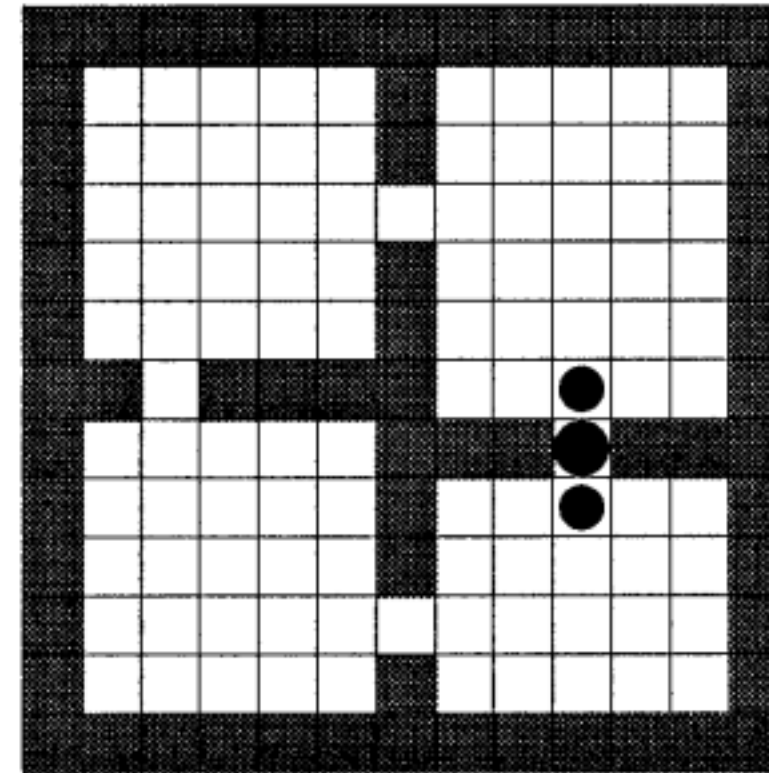
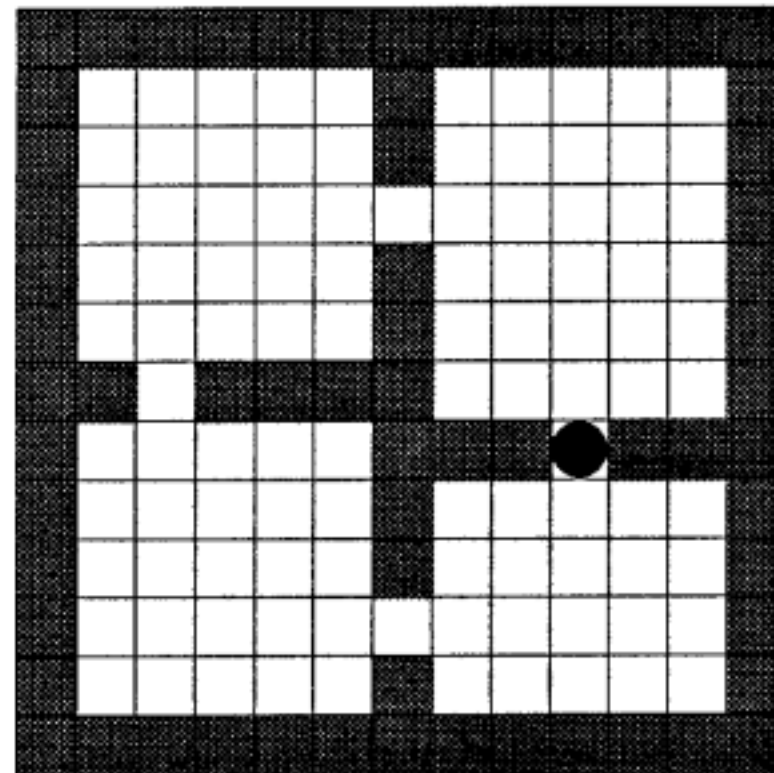
*4 stochastic  
primitive actions*



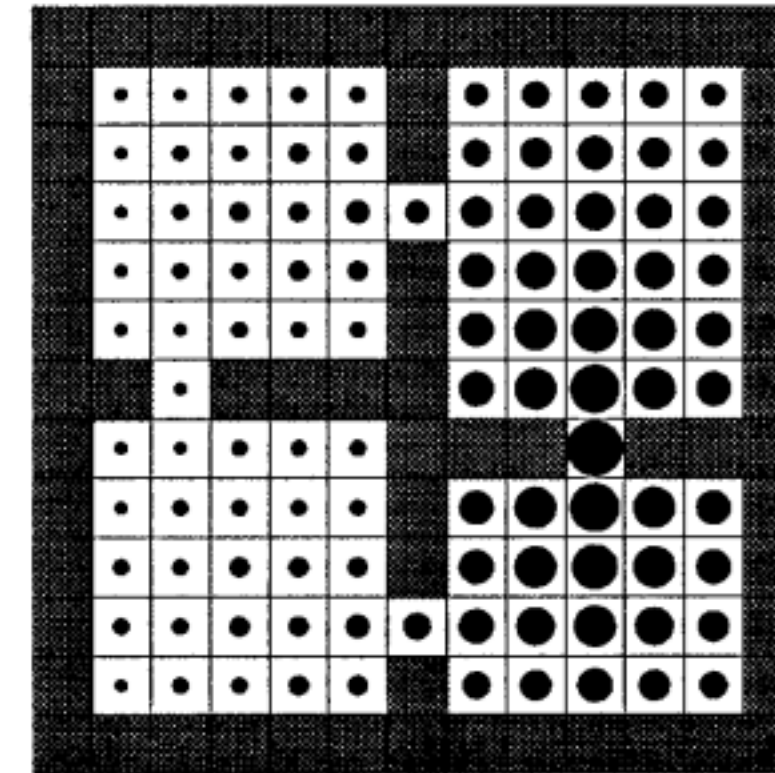
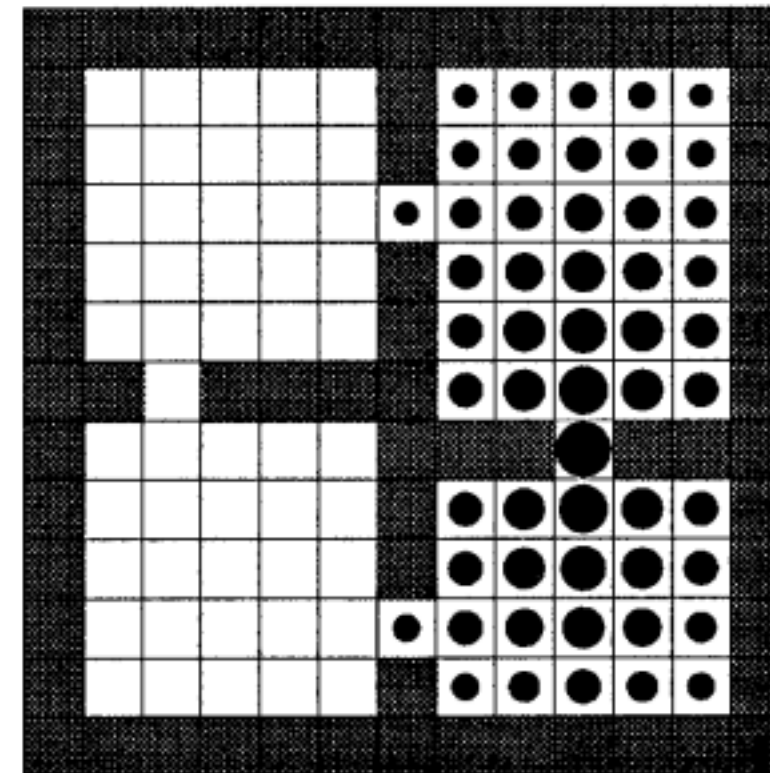
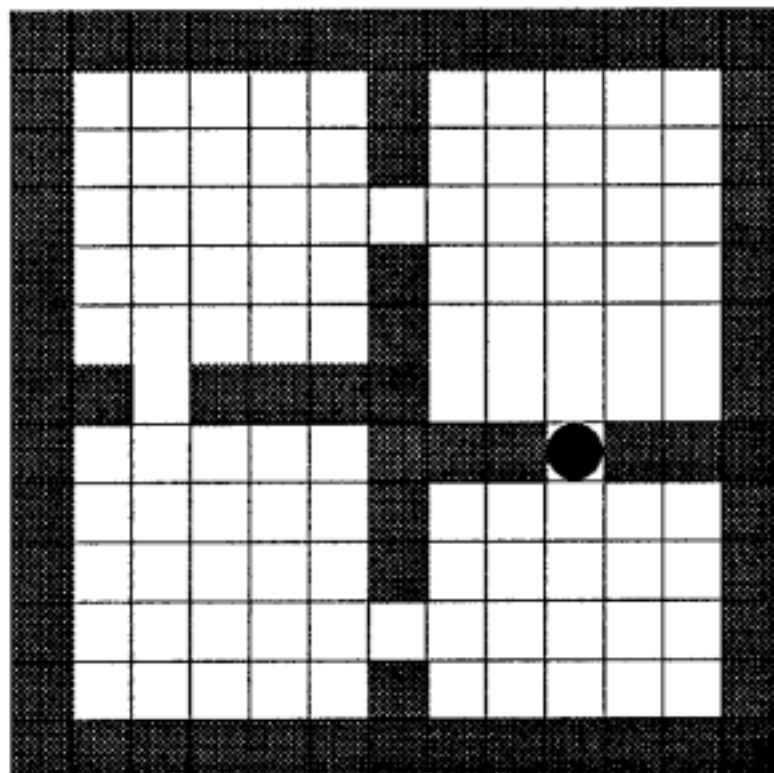
*8 multi-step options  
(to each room's 2 hallways)*

# Examples

Primitive  
options  
 $\mathcal{O}=\mathcal{A}$



Hallway  
options  
 $\mathcal{O}=\mathcal{H}$



Initial Values

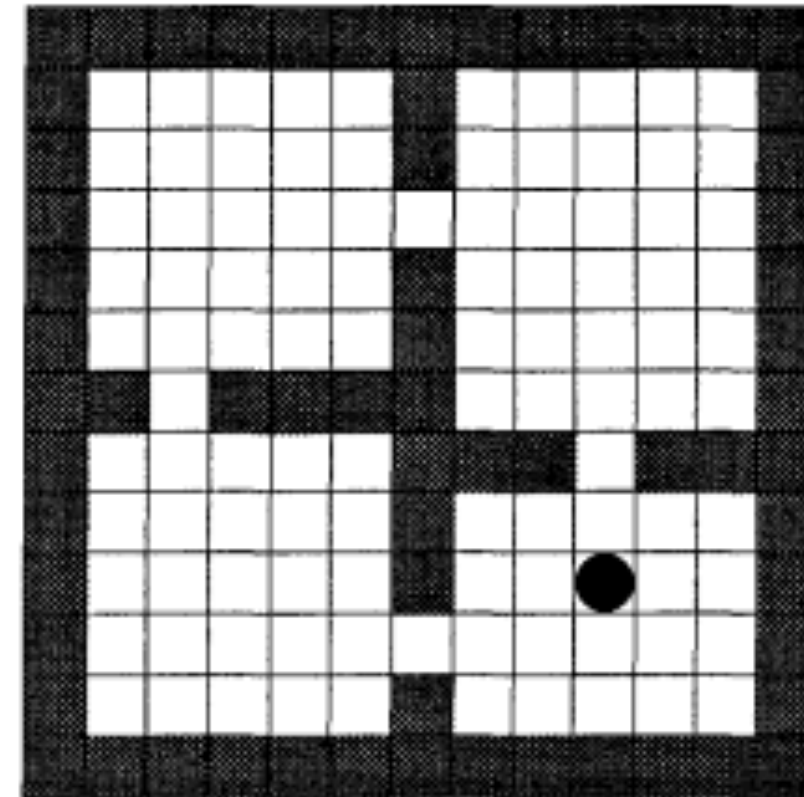
Iteration #1

Iteration #2

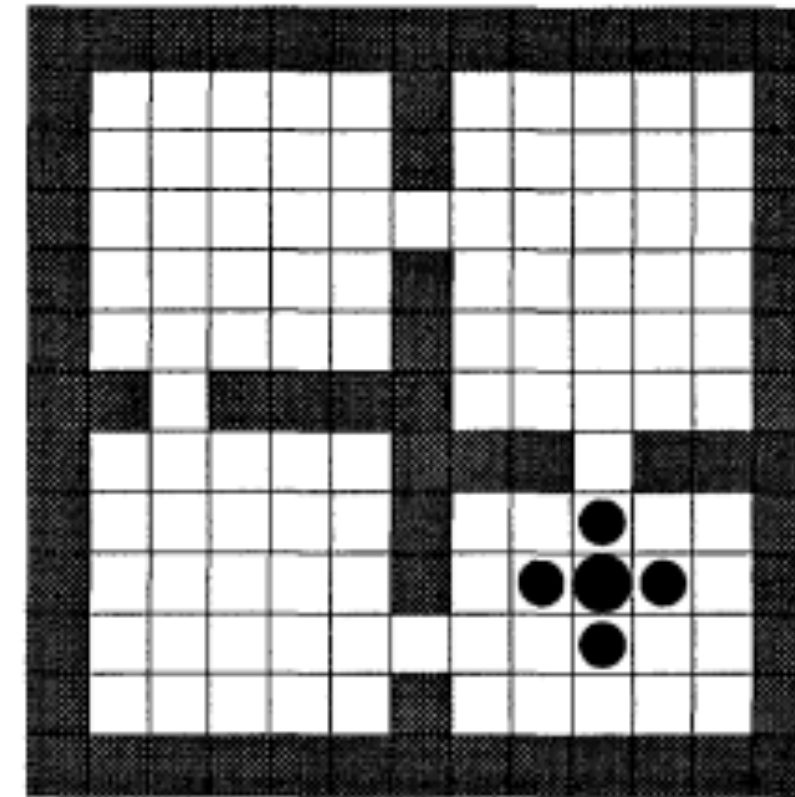
# Examples

Primitive  
and  
halfway  
options

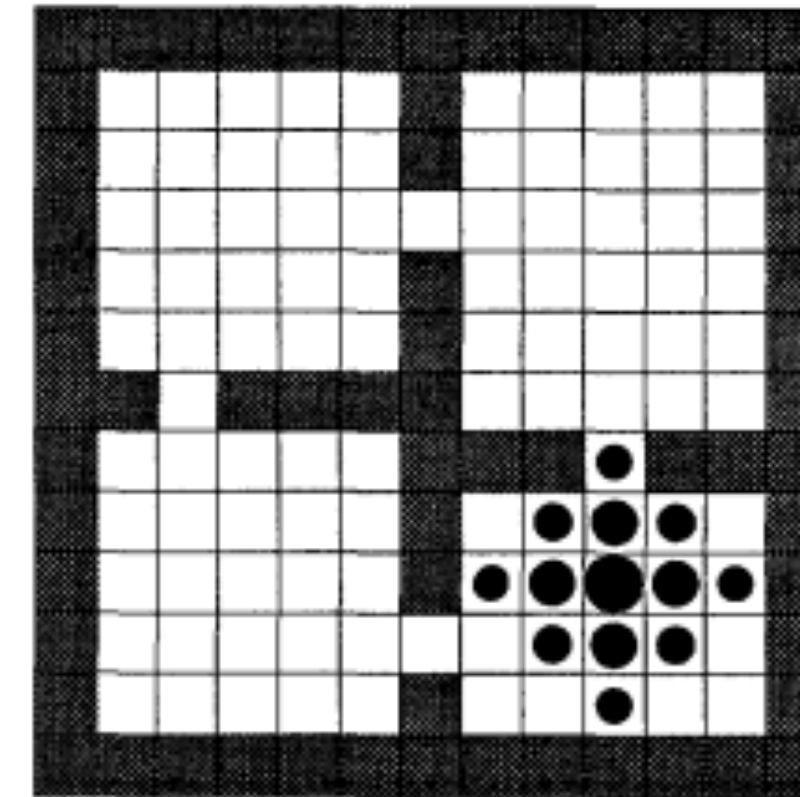
$$O = AUH$$



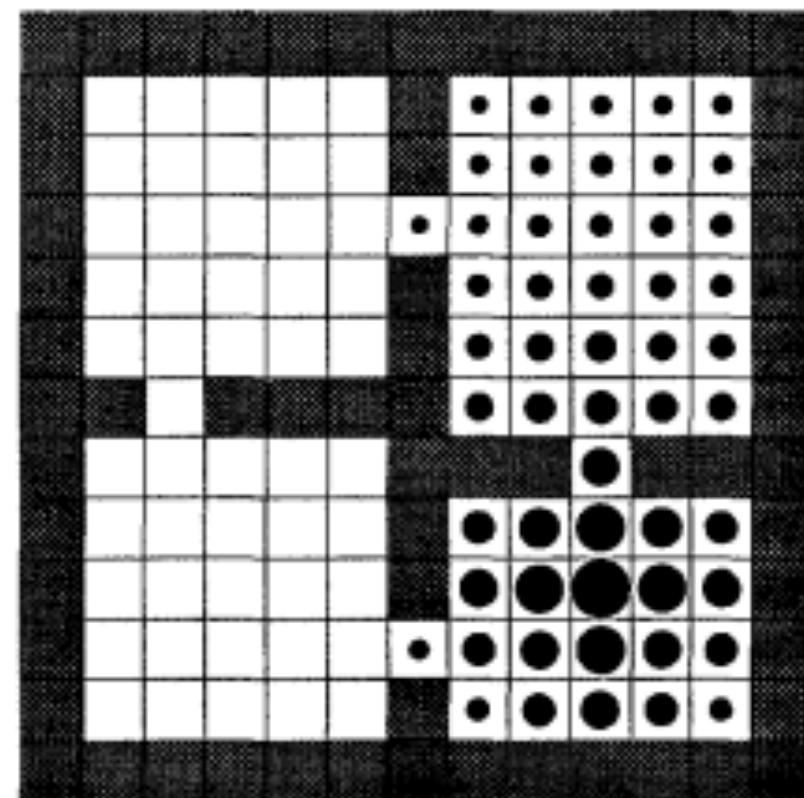
Initial values



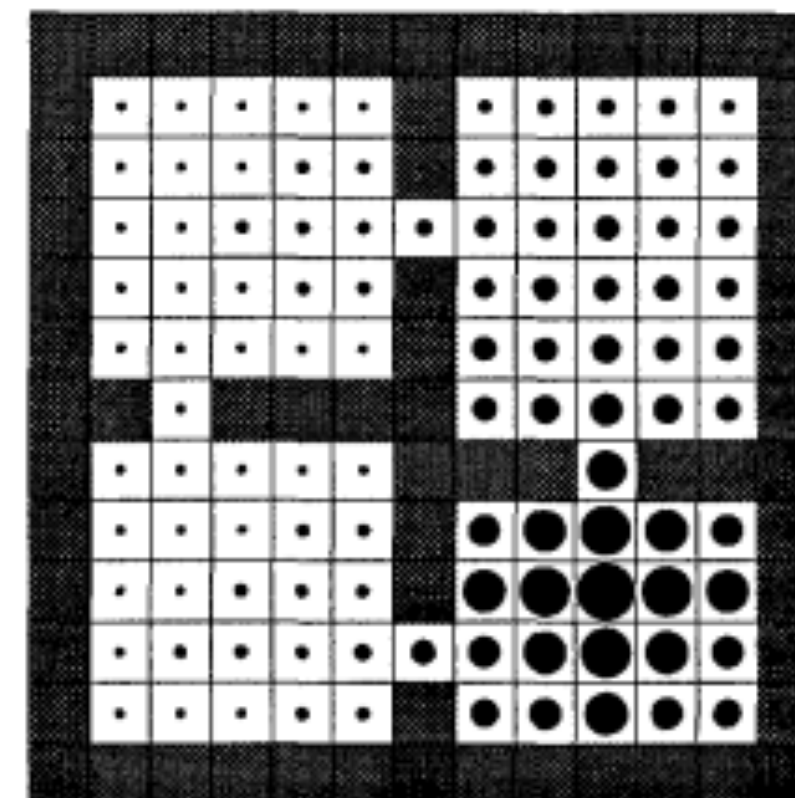
Iteration #1



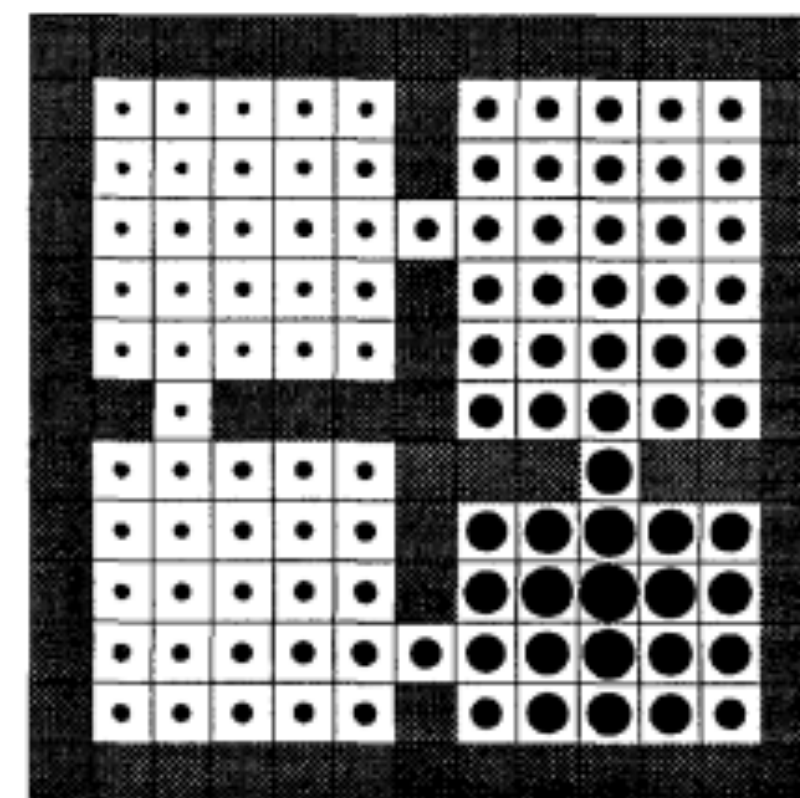
Iteration #2



Iteration #3



Iteration #4



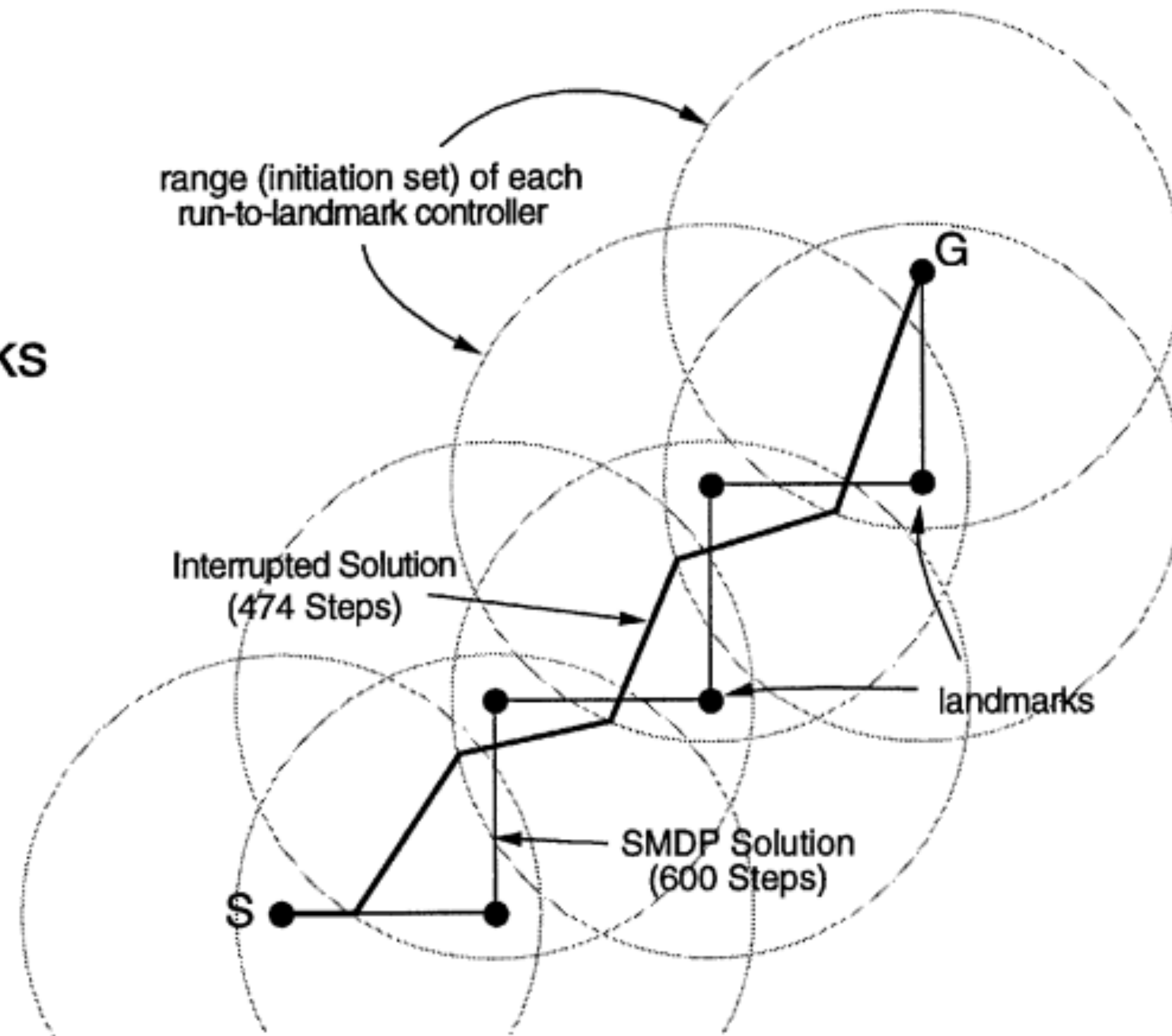
Iteration #5

# Intra-Option Learning

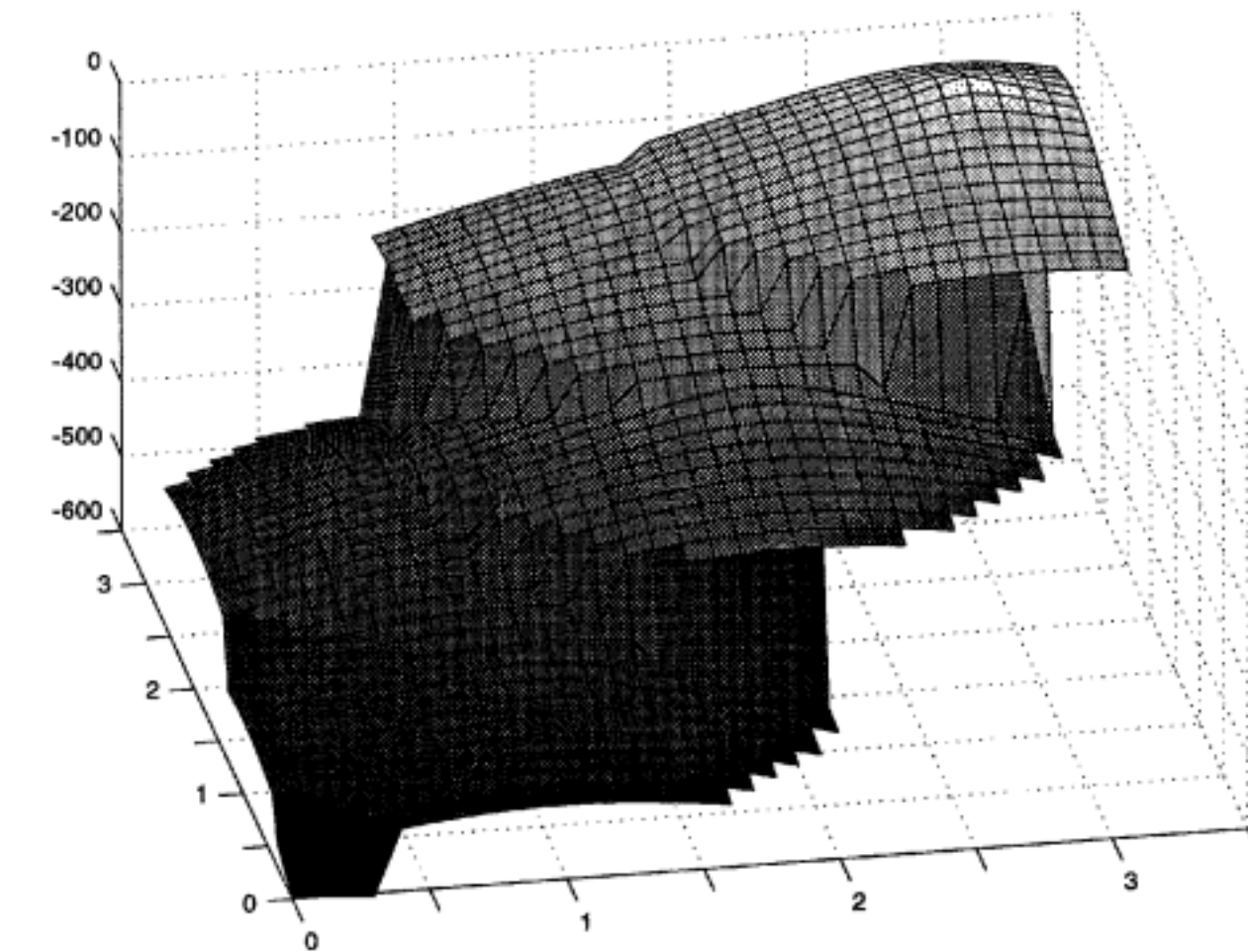
- Key benefit of the options framework over semi-MDPs is the ability to “open up” and modify options.
- Interrupt an option before termination if another option has higher predicted value (Theorem 2 of reading guarantees improvement).
- Learn about or improve options:
  - Build multi-time models for planning (section 5).
  - Learn option-values more efficiently (section 6).
  - Improve option policy (section 7).

# Interrupting Options

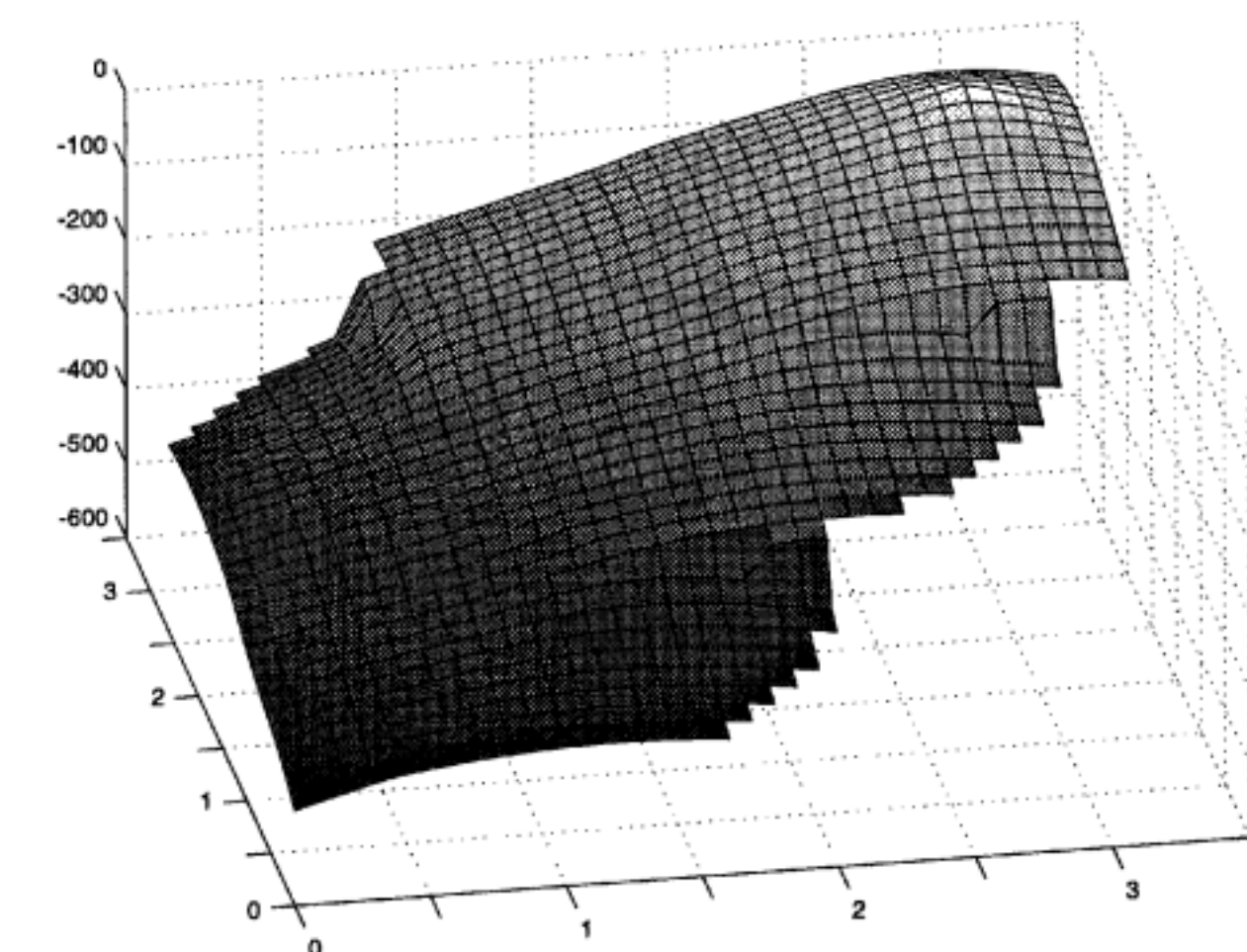
## Landmarks Problem



$V_O^*$ - SMDP Value Function



$V^{\mu'}$ - Values with Interruption



# Where do options come from?

- Manually defined based on domain knowledge.
- Learned from experience
  - Can specify sub-goal rewards and learn an option (i.e., a policy) that maximizes sub-goal reward.
  - Option-Critic architecture learns options based on original problem reward.

# Apoorva's Presentation

- The Option Critic Architecture.
- Bacon et al. 2018
- Slides

# Options Today

- The options framework is perhaps the most well-known formalism for hierarchical RL.
  - Not the only choice!
- Alternative skill formalism in some works:
  - $\pi_{\text{high}}(z | s)$  and  $\pi_{\text{low}}(a | s, z)$ .
- Intra-option learning and reasonable sub-goals are still open challenges with no widely accepted one method.

# Summary

- Temporal abstraction facilitates exploration and credit-assignment.
- Options framework enables extending concepts, theory, and algorithms from MDPs to temporal abstraction.
- Key questions:
  - How are options acquired?
  - How much benefit can we derive from first learning options and then learning the optimal policy over options?

# Action Items

- Study for midterm exam!
- Continue working on final projects.
- Read Chapter 1 and 5 of “Multi-agent RL: Foundations and Modern Approaches”