Lecture 12: Options

B. Ravindran

Options Framework

Options (Sutton, Precup, & Singh, 1999): A generalization of actions to include temporally-extended courses of action

An option is a triple $o = < I, \pi_o, \beta >$

- $I \subseteq S$ is the set of states in which o may be started
- $\pi_o : \Psi \to [0,1]$ is the (stochastic) policy followed during o
- $\beta : S \to [0,1]$ is the probability of terminating in each state

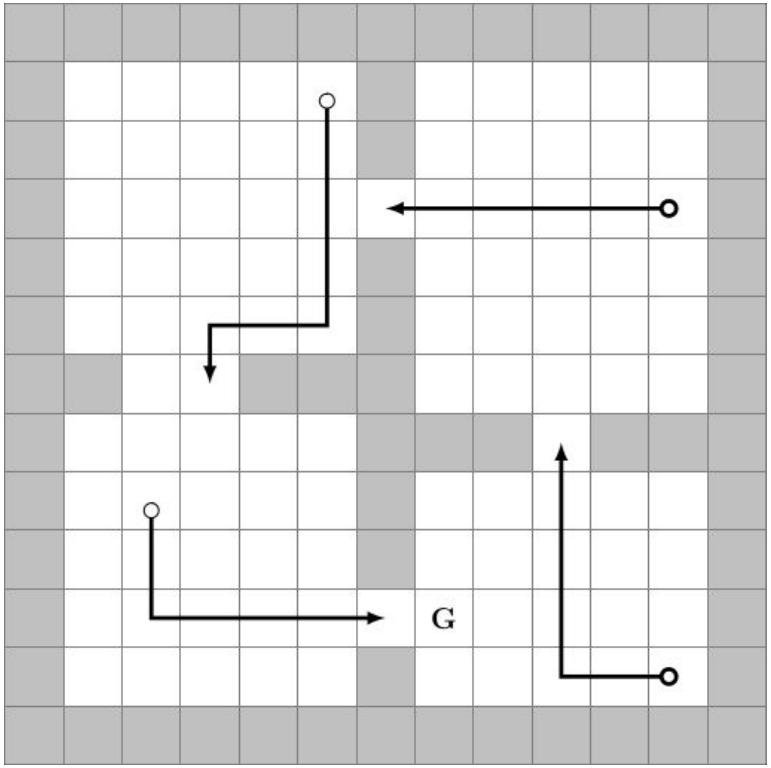
Generalising over Tasks

- Each task has a different reward structure in the state space

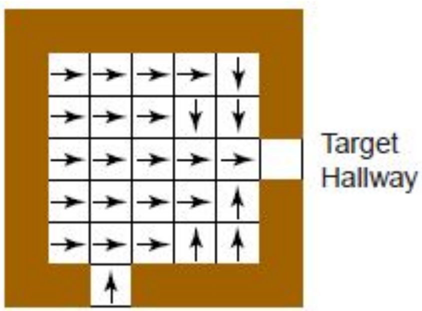
- Options provide a model for subtasks

- Semi-Markov Processes

- Can use generalization of TD, Q-learning, SARSA, etc. with options



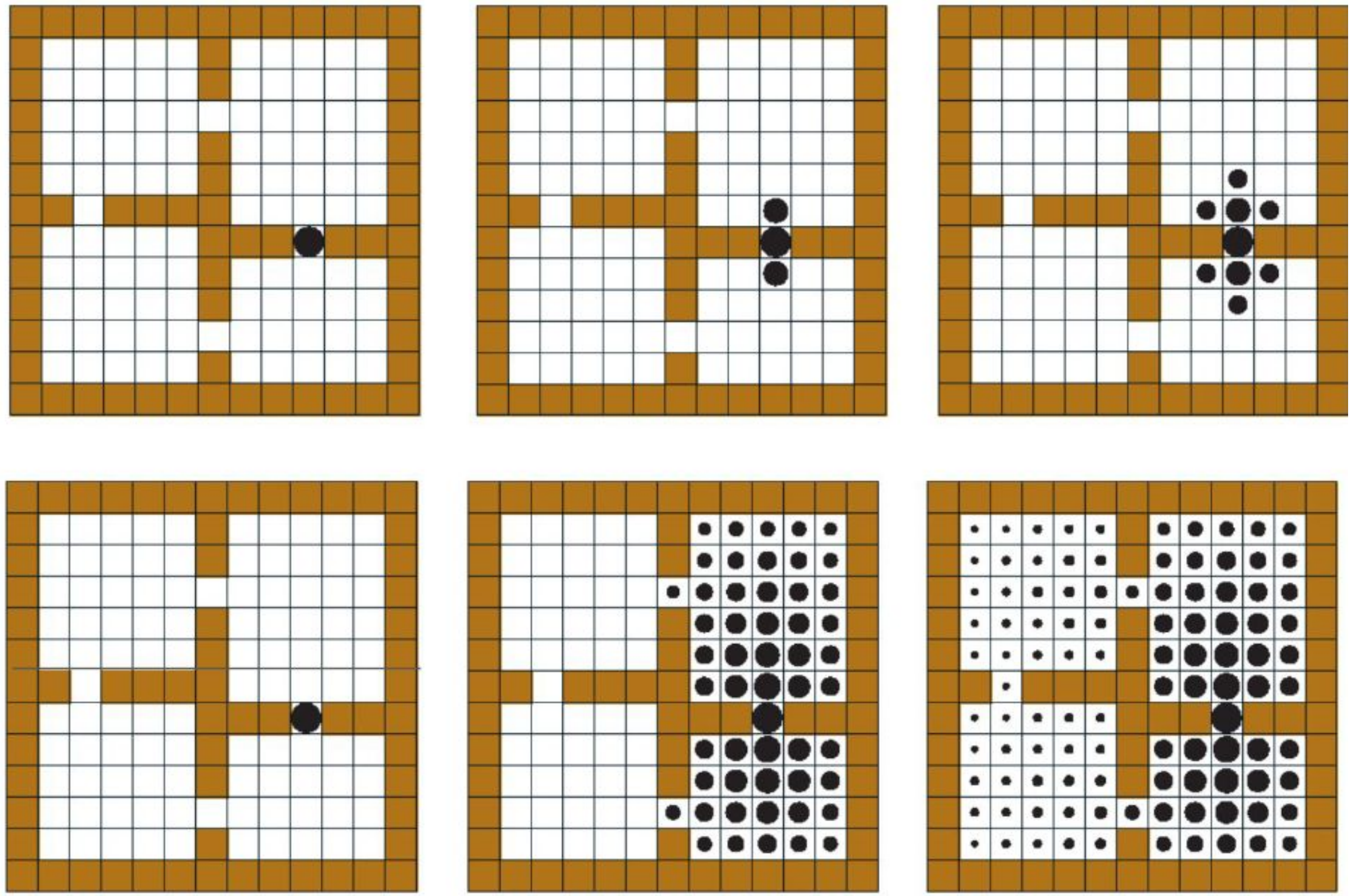
Speedup using Options



Underlying policy of one hallway option

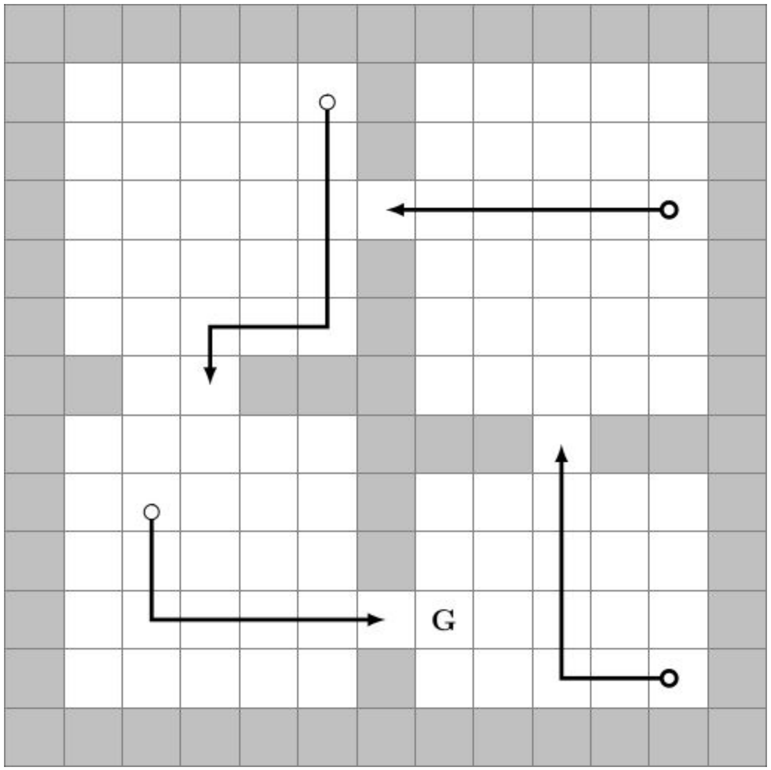
Primitive Actions

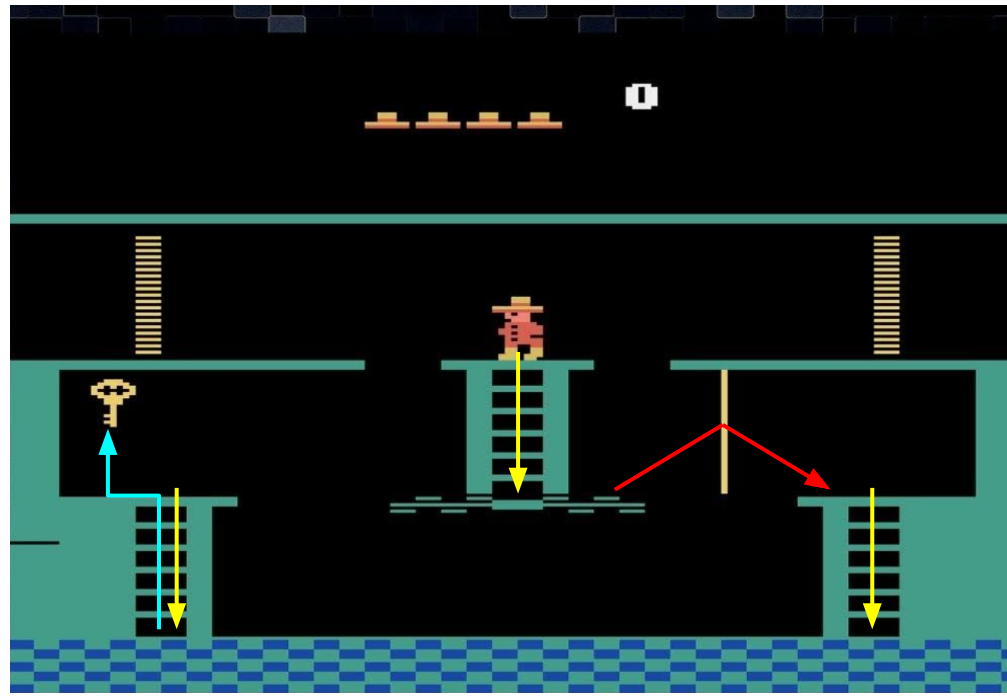
Hallway Options



Initial Values | Iteration # 1 | Iteration # 2

Sub-goal Options

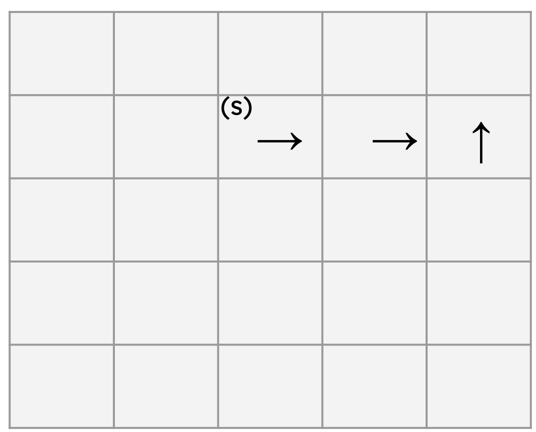




Types of Options

- Markov Options:
  - π_o depends only on current state

- Semi-Markov Options:
  - π_o depends on history since option started



Learning with Options

SMDP Q-Learning: In a *state s*,

- If a primitive *action a* is selected, *Q(s,a)* is updated according to the regular Q-Learning update rule

- If an *option o* is selected, no state-action values are updated until o terminates
- The cumulative, discounted reward received during the execution of *option o* is used to update only *Q(s,o)*

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha\left[\bar{r}_{t+\tau} + \gamma^{\tau} \max_{a'} Q\big(s_{t+\tau}, a'\big) - Q(s_t, a_t)\right]$$

$$\bar{r}_{t+\tau} = r_{t+1} + \gamma\, r_{t+2} + \ldots + \gamma^{\tau-1} r_{t+\tau}$$

Intra-Option Q-Learning

- At every step, the state-action value of the
  - primitive action, as well as
  - options that would have selected the same action

  are updated, regardless of the option in effect

For $\pi_o - a_1(for\ s_1)\,, a_2\,(for\ s2)\ldots\ldots$

Option o is updated as

$$
\begin{aligned}
Q(s_1, o) &= Q(s_1, o) + \alpha[r_1 + \gamma Q(s_2, o) - Q(s_1, o)] \qquad \text{If not terminating at } s_2 \\
&= Q(s_1, o) + \alpha\Big[r_1 + \gamma \max_a Q(s_2, a) - Q(s_1, o)\Big] \; \text{If terminating at } s_2
\end{aligned}
$$

Intra-Option Q-Learning

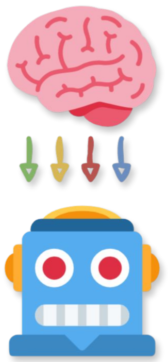
- For primitive actions (state-action pairs), we use regular Q-learning update

$$Q(s_1, a_1) = Q(s_1, a_1) + \alpha\Big[r_1 + \gamma \max_a Q(s_1, a) - Q(s_1, a_1)\Big]$$
$$Q(s_2, a_2) = \ldots$$

- Additionally, an option execution allows us to update for all other options that are consistent with the first option *(every other option o' where the same action would have been selected)*

- Suppose $\pi_o(s_1) = a \quad \& \quad \pi_{o'}(s_1) = a$
- When executing option o, option o' can also be updated

$$Q\left(s_1, o'\right) = Q\left(s_1, o'\right) + \alpha\left(r_1 + \gamma Q\left(s_2, o'\right) - Q\left(s_1, o'\right)\right)$$



Discovering Options

Goodness Measures

- What are good options?
  - Connect to bottlenecks
  - Speed up learning; Useful for a family of tasks
    - Essentially transfer
  - Reduce uncertainty
  - Explainability
  - Mainly heuristics
    - Not connected directly to performance on MDPs

Discovering Options

- Bottlenecks [McGovern & Barto, Stolle & Precup, etc.]
- Graph partitions [Mannor et al.]
- Betweenness [Simsek & Barto]
- Frequency of changes [Jonsson & Barto, Hengst]
- Bisimulation metrics [Castro & Precup]
- Intrinsic Motivation [Satinder et al.; Barto et al.]

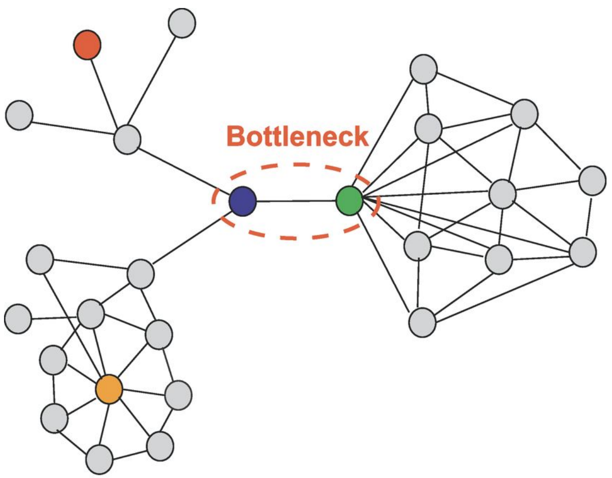
- Our Contribution: Metastability, Small World Options, etc.

Finding Bottlenecks

- MDP can be segmented and modeled as a graph (Nodes = states; Edge = action)

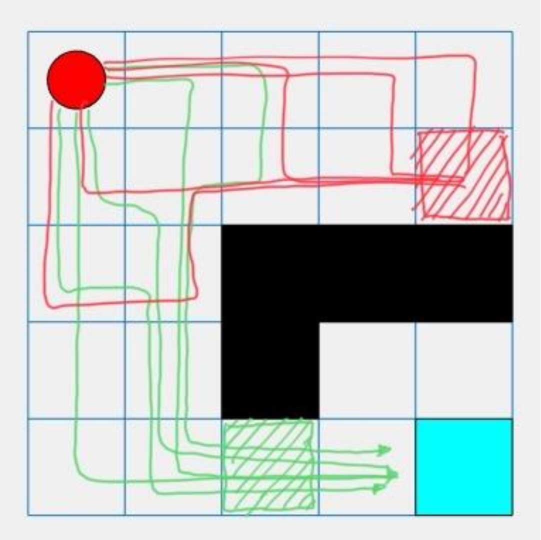
- Find components of graph which are weakly connected (Graph partitioning)

- States where weak connections happen are 'bottleneck' states



Diverse Density

- ❑ Assume we've obtained a bunch of successful & unsuccessful trajectories (Experience Replay)
- ❑ Find states that appear frequently on successful and rarely on unsuccessful trajectories
- ❑ "The agent needs to get through these states to reach the goal"

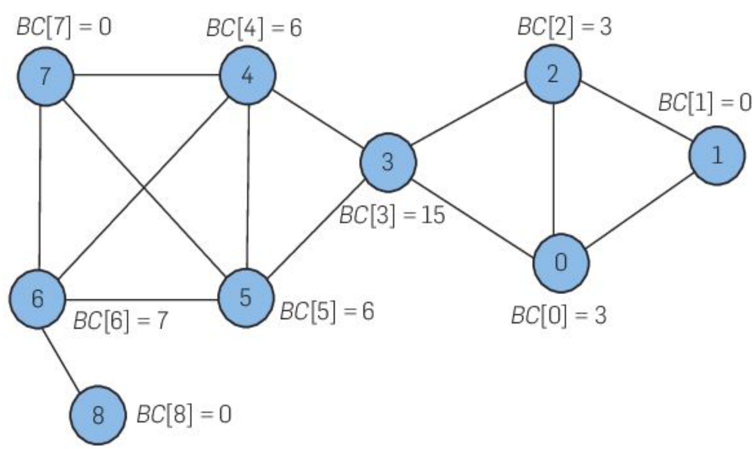


RED : unsuccessful trajectories

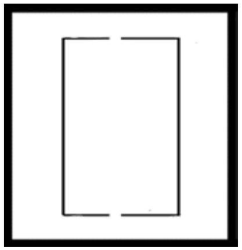
GREEN : successful trajectories

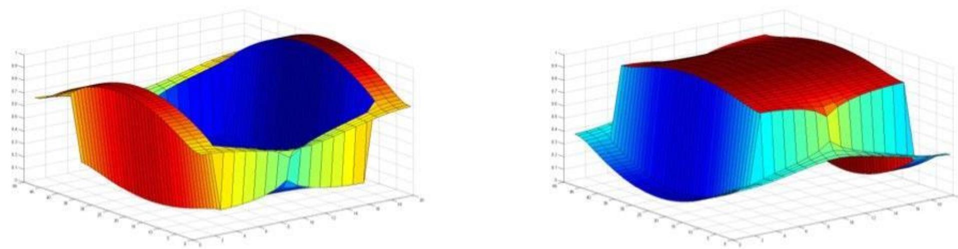
Betweenness Centrality

- Pick (all possible) pair of nodes on the graph and calculate the shortest path
- A node has high betweenness if many shortest-paths pass through it
- States with high betweenness centrality can be considered bottlenecks

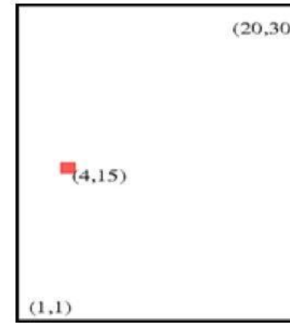


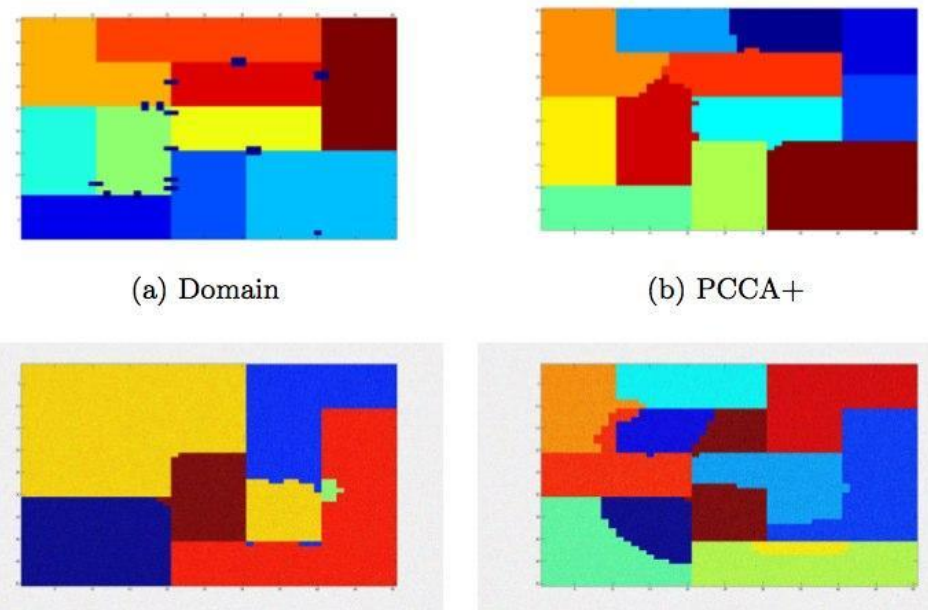
Metastable Regions





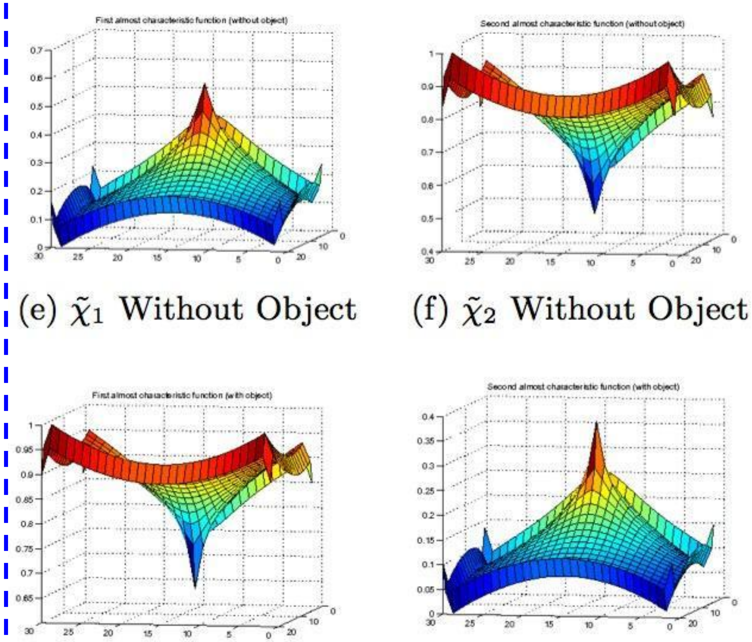
(a) Inner (b) Outer





(a) Domain (b) PCCA+

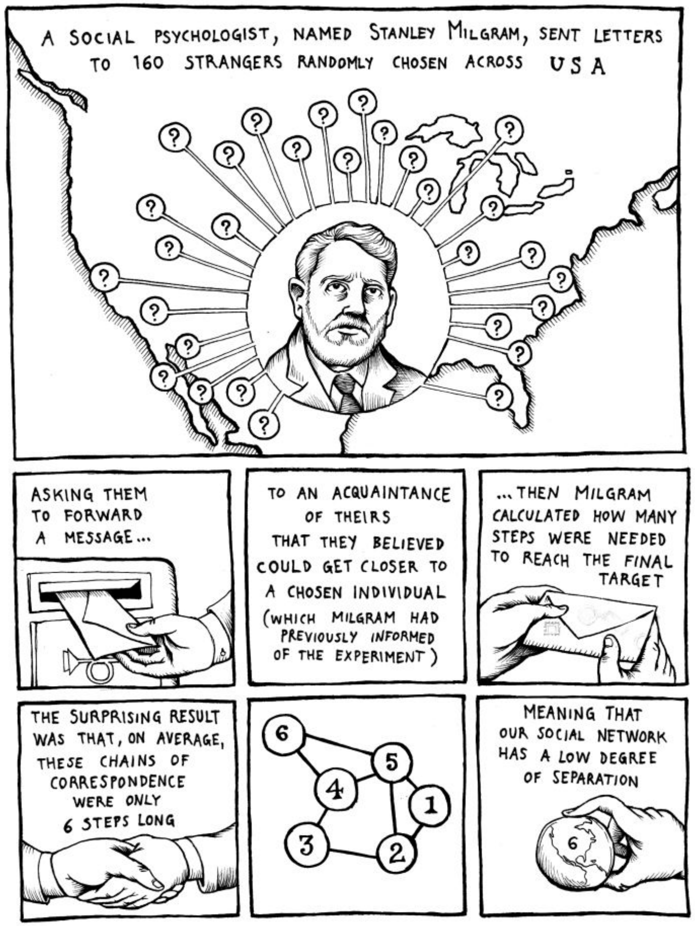
(c) Kannan (d) NCut



(e) $\tilde{\chi}_1$ Without Object (f) $\tilde{\chi}_2$ Without Object

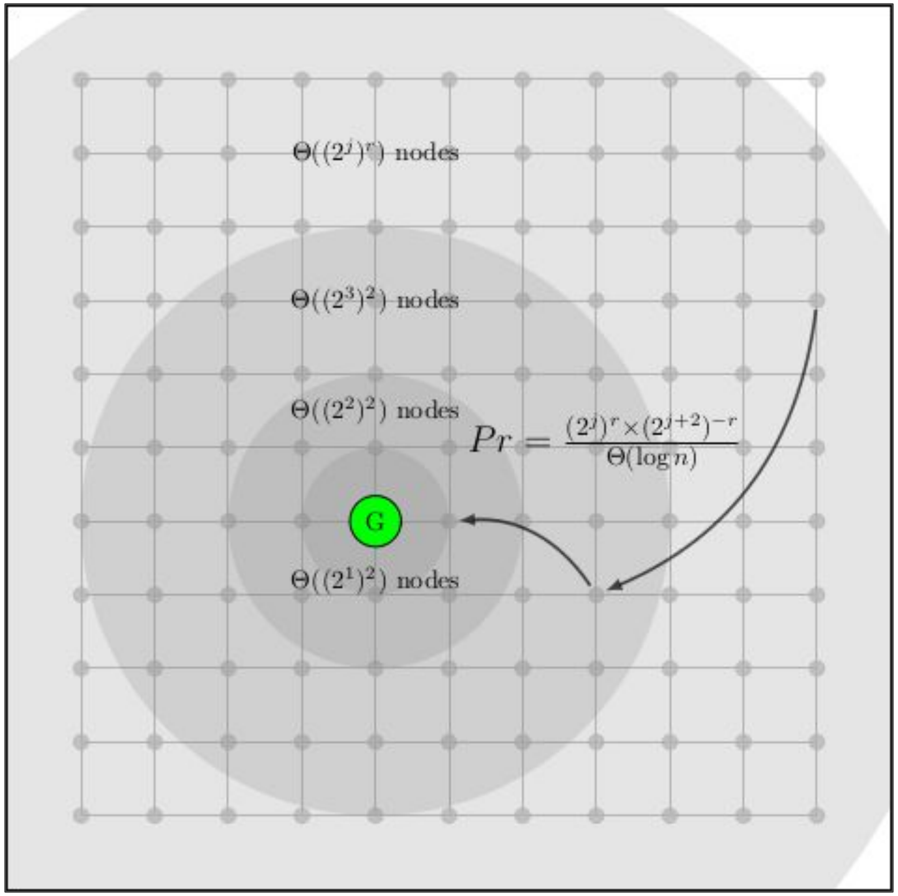
(g) $\tilde{\chi}_1$ With Object (h) $\tilde{\chi}_2$ With Object

Small World Options



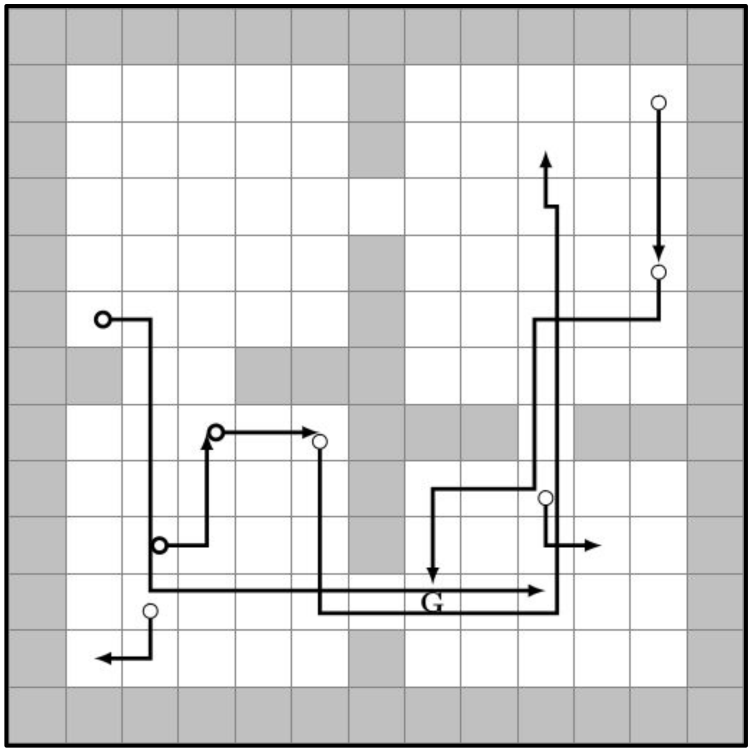
- ❑ Random options inserted with the expected length of these options following a certain probability distribution
  - ❑ Inspired by Stanley Milgram's experiment
- ❑ Exploration time can be cut down significantly
- ❑ Makes best use of data in comparison

Small World Options



- A r-dimensional lattice graph
- Edges distributed inversely proportional to distance
- A greedy agent will move from one neighbourhood to another in log(n) time

Small Worlds in RL



- ❑ Construct “path options” that take an agent from state s to s’
- ❑ s′ is chosen according to the power-law
- ❑ Which distance based?
- ❑ Value and state-space distance are related