Announcements

- Mid-term exam
  - Closed-book, no electronic device
  - 2 A4 cheatsheets (i.e., 4 pages in total), handwritten (your own writing), stapled, photocopy not allowed
- Nov. 4 class moved to Nov. 7 10-11:40am (D205)
- Mid-term course evaluation
- OH today 1pm-2pm

Ve492: Introduction to Artificial Intelligence
Mid-term Review

Paul Weng

UM-SJTU Joint Institute

Slides adapted from http://ai.berkeley.edu, AIMA, UM, CMU

Quiz: Search

- Consider a graph search problem where for every action, the cost is at least ϵ, with $\epsilon>0$. Assume the used heuristic is consistent.
  - Depth-first graph search is guaranteed to return an optimal solution.
  - Breadth-first graph search is guaranteed to return an optimal solution.
  - Uniform-cost graph search is guaranteed to return an optimal solution.
  - Greedy graph search is guaranteed to return an optimal solution.
  - A* graph search is guaranteed to return an optimal solution.
  - A* graph search is guaranteed to expand no more nodes than depth-first graph search.
  - A* graph search is guaranteed to expand no more nodes than uniform-cost graph search.

Quiz: A*

- Assume we are running A* graph search with a consistent heuristic h. Assume the optimal cost path to reach a goal has a cost c^*.
  - All nodes n reachable from the start state satisfying $g(n) < c^*$ will be expanded during the search
  - All nodes n reachable from the start state satisfying $f(n) = g(n) + h(n) < c^*$ will be expanded during the search
  - All nodes n reachable from the start state satisfying $h(n) < c^*$ will be expanded during the search

Quiz: A* heuristics

- Let $h_1(s)$ be an admissible heuristic. Let $h_2(s) = 2h_1(s)$.
  - The solution found by A* tree search with h_2 is guaranteed to be an optimal solution.
  - The solution found by A* tree search with h_2 is guaranteed to have a cost at most twice as much as the optimal path.
  - The solution found by A* graph search with h_2 is guaranteed to be an optimal solution.

Quiz: A* Heuristics

- Let H_1 and H_2 both be admissible heuristics.
  - $\max(H_1, H_2)$ is necessarily admissible
  - $\min(H_1, H_2)$ is necessarily admissible
  - $(H_1 + H_2)/2$ is necessarily admissible
  - $\max(H_1, H_2)$ is necessarily consistent

Quiz: A* Heuristics

- Let H_1 be an admissible heuristic, and let H_2 be an inadmissible heuristic.
  - $\max(H_1, H_2)$ is necessarily admissible
  - $\min(H_1, H_2)$ is necessarily admissible
  - $(H_1 + H_2)/2$ is necessarily admissible
  - $\max(H_1, H_2)$ is necessarily consistent

Quiz: Search under Uncertainty

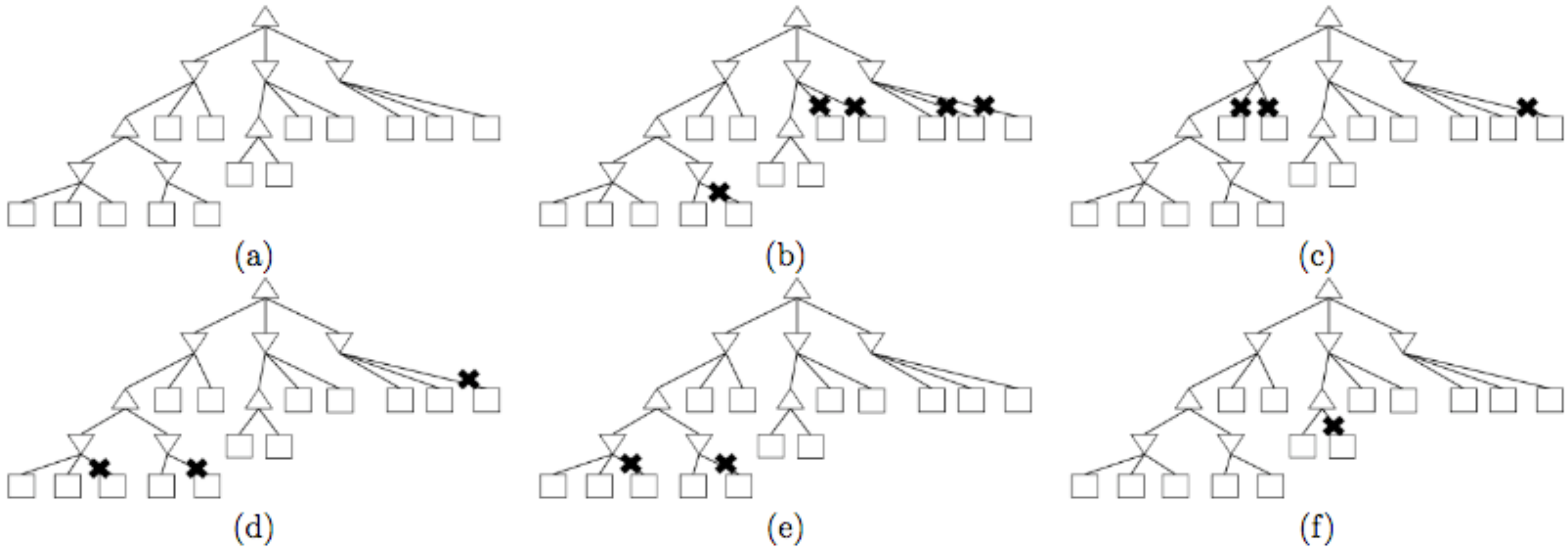
- You are given a game tree for which you are the maximizer, and in the nodes in which you don't get to make a decision an action is chosen uniformly at random amongst the available options. Your objective is to maximize the probability you win $10 or more (rather than the usual objective to maximize your expected value).
  - Running expectimax will result in finding the optimal strategy to maximize the probability of winning $10 or more.
  - Running minimax, where chance nodes are considered minimizers, will result in finding the optimal strategy to maximize the probability of winning $10 or more.
  - Running expectimax in a modified game tree where every pay-off of $10 or more is given a value of 1, and every pay-off lower than $10 is given a value of 0 will result in finding the optimal strategy to maximize the probability of winning $10 or more.
  - Running minimax in a modified game tree where every pay-off of $10 or more is given a value of 1, and every pay-off lower than $10 is given a value of 0 will result in finding the optimal strategy to maximize the probability of winning $10 or more.

Quiz: Adversarial Search

- In the context of adversarial search, α-β pruning
  - can reduce computation time by pruning portions of the game tree
  - is generally faster than minimax, but loses the guarantee of optimality
  - always returns the same value as minimax for the root of the tree
  - always returns the same value as minimax for all nodes on the leftmost edge of the tree, assuming successor game states are expanded from left to right
  - always returns the same value as minimax for all nodes of the tree

Quiz: Adversarial Search

- Assume we run α–β pruning expanding successors from left to right on a game with tree as shown in Figure (a).



(a) (b) (c) (d) (e) (f)

  - For some choice of pay-off values, no pruning will be achieved (shown in Figure (a)).
  - For some choice of pay-off values, the pruning shown in Figure (b) will be achieved.
  - For some choice of pay-off values, the pruning shown in Figure (c) will be achieved.
  - For some choice of pay-off values, the pruning shown in Figure (d) will be achieved.
  - For some choice of pay-off values, the pruning shown in Figure (e) will be achieved.
  - For some choice of pay-off values, the pruning shown in Figure (f) will be achieved.

Quiz: MDP

- For Markov Decisions Processes (MDPs), we have that:
  - A small discount (close to 0) encourages shortsighted, greedy behavior.
  - A large, negative living reward ($\ll 0$) encourages shortsighted, greedy behavior.
  - A negative living reward can always be expressed using a discount<1.
  - A discount<1 can always be expressed as a negative living reward.

Quiz: MDP

- Assume given an MDP $\mathscr{M} = (S, A, T, R, \gamma)$. Define a new MDP $\mathscr{M}' = (S, A, T, \alpha R + \beta, \gamma)$ where $\alpha > 0$ and $\beta \in \mathbb{R}$.
  - An optimal policy in $\mathscr{M}$ is optimal in $\mathscr{M}'$.
  - An optimal policy in $\mathscr{M}'$ is optimal in $\mathscr{M}$.
  - If $\pi \gtrsim \pi'$ in $\mathscr{M}$, then $\pi \gtrsim \pi'$ in $\mathscr{M}'$
  - If $\pi \gtrsim \pi'$ in $\mathscr{M}'$, then $\pi \gtrsim \pi'$ in $\mathscr{M}$

Quiz: MDP

- Value iteration can converge only if the discount factor (γ) satisfies $0 < \gamma < 1$.
- Policies found by value iteration may be superior to policies found by policy iteration.
- Policies found by policy iteration may be superior to policies found by value iteration.

Quiz: Reinforcement Learning

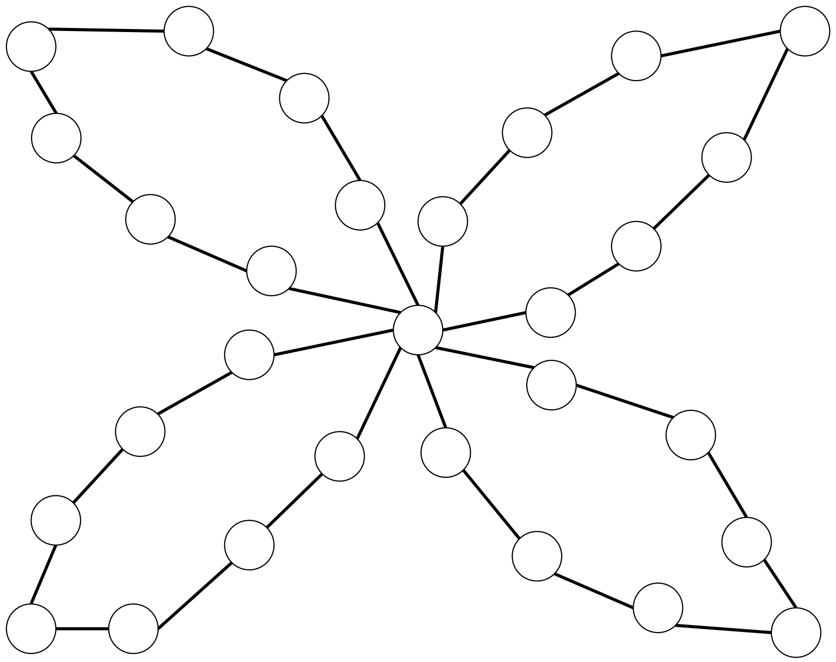
- Assume that the agent observes the true reward with some Gaussian noise $\mathcal{N}(0,1)$, Q-learning would still converge
- Q-learning can learn the optimal Q-function Q^* without ever executing the optimal policy.
- If an MDP has a transition model T that assigns non-zero probability for all triples T (s, a, s′) then Q-learning will fail.
- In Q-learning, we decide to explore every k steps, i.e., if t = 0 [k] we choose a random action with a uniform distribution, otherwise we choose the greedy action. This version would still converge.

Quiz: CSP

- The most-constrained variable heuristic provides a way to select the next variable to assign in a backtracking search for solving a CSP.
- By using the most-constrained variable heuristic and the least-constraining value heuristic we can solve every CSP in time linear in the number of variables.
- CSP problems are always solved faster with arc consistency than with forward checking.
- When enforcing arc consistency, the values that remain in each domain depend on the order in which the arcs are considered.

Quiz: CSP

- Assume given a CSP whose constraint graph is given below and that all the variables have the same domain.
- What is the complexity of solving it with a direct application of backtracking search?
- Which efficient strategy could you apply to solve it? What would be the complexity?



CSP Problem: Job Scheduling

- When can I move in?

Task	Description	Duration	Predecessor
a	Erecting walls	7	none
b	Carpentry for roof	3	a
c	Roof	1	b
d	Installations	8	a
e	Facade painting	2	c & d
f	Windows	1	c & d
g	Garden	1	c & d
h	Ceilings	3	a
i	Painting	2	f & h
j	Moving in	1	i

MDP Problem: Post-Decision State

- Consider an infinite-horizon, discounted MDP (S, A, T, R, γ) where T(s,a,s′) = P(s′ | f(s,a)), R(s,a,s′) = R(s,a) and f is some deterministic function mapping $S \times A \rightarrow Y$, where Y is a set of states called post-decision states.

$$
\begin{tikzcd}
(s_0, a_0) \arrow[r, "f"] \arrow[dr, dotted] & y_0 \arrow[r, "P"] & (s_1, a_1) \arrow[r, "f"] \arrow[dr, dotted] & y_1 \arrow[r, "P"] & (s_2, a_2) \arrow[r, "f"] \arrow[dr, dotted] & \cdots \\
& r_0 & & r_1 & & r_2
\end{tikzcd}
$$

- $V^\pi(s_0) = \mathbb{E}[R(s_0, \pi(s_0)) + \gamma R(s_1, \pi(s_1)) + \gamma^2 R(s_2, \pi(s_2)) + \ldots]$
- $W^\pi(y_0) = \mathbb{E}[R(s_1, \pi(s_1)) + \gamma R(s_2, \pi(s_2)) + \gamma^2 R(s_3, \pi(s_3)) + \ldots]$

MDP Problem: Post-Decision State

- Write W^* in terms of V^*
- Write V^* in terms of W^*
- Provide the equivalent of the Bellman equation for W^*
- Fill in the blanks
  - Initialize policy π_1 arbitrarily
  - For all $i = 1,2,\ldots$
    - Compute $W^{\pi_i}(y) \quad \forall y$
    - Compute new policy π_{i+1} from $\pi_{i+1}(s) = arg\max\_\_\_\_\_\_ \quad \forall s$
    - If $\_\_\_\_\_\_ \quad \forall s$, return π_i

Game Theory Problem

- Two players choose simultaneously a coin of 10 cents, 50 cents or 1 dollar, which they show to each other.
- If they chose the same coin, player I wins. Otherwise, player II wins.
- Write this game in normal form. Is there any pure NE?
- Express a system of inequalities to find a mixed NE.

Search: The wolf, goat, cabbage problem

- You are on the bank of a river with a boat, a cabbage, a goat, and a wolf. Your task is to get everything to the other side. Rules:
  - Only you can handle the boat
  - When you're in the boat, there is only space for one more item
  - You can't leave the goat alone with the wolf, nor with the cabbage (or something will be eaten)