CS 188
Summer 2023

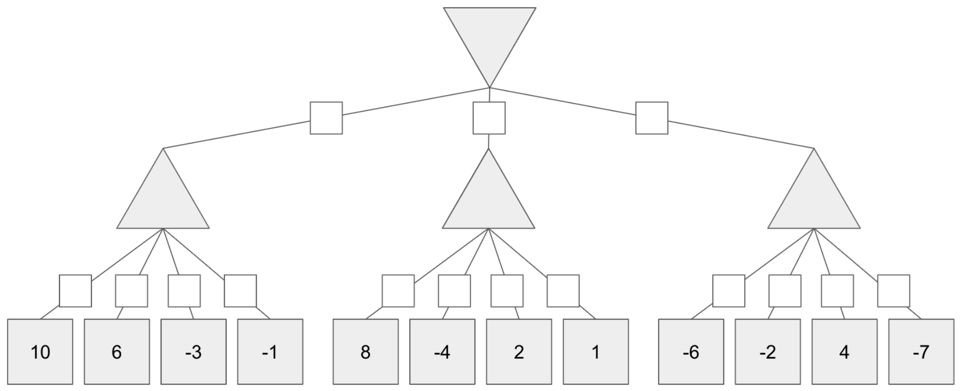
Final Review Games Solutions

Q1. Coin Stars

In a new online game called Coin Stars, all players are walking around an M x N grid to collect **hidden coins, which only appear when you're on top of them**. There are also power pellets scattered across the board, which are visible to all players. If you walk onto a square with a power pellet, your power level goes up by 1, and the power pellet disappears. Players will also attack each other if one player enters a square occupied by another player. In an attack, the player with a higher power level will steal all the coins from the other player. If they have equal power levels, nothing happens. Each turn, players go in order to move in one of the following directions: {N, S, E, W}.

In this problem, you and your friend Amy are playing Coin Stars against each other. You are player 1, and your opponent Amy is player 2. Our state space representation includes the locations of the power pellets (x_{p_j}, y_{p_j}) and the following player information: (1) Each player's location (x_i, y_i); (2) Each player's power level l_i; (3) Each player's coin count c_i.

(a) Suppose a player wins by collecting more coins at the end of a number of rounds, so we can formulate this as a minimax problem with the value of the node being $c_1 - c_2$. Consider the following game tree where you are the maximizing player (maximizing the your net advantage, as seen above) and the opponent is the minimizer. Assuming both players act optimally, if a branch can be pruned, fill in its square completely, otherwise leave the square unmarked.



● None of the above can be pruned

If you traverse the tree with $\alpha - \beta$ pruning, you will see that no branches can be pruned

(b) Suppose that instead of the player with more coins winning, every player receives payout equal to the number of coins they've collected. Can we still use a multi-layer minimax tree (like the one above) to find the optimal action?

○ Yes, because the update in payout policy does not affect the minimax structure of the game.

○ Yes, but not for the reason above

○ No, because we can no longer model the game under the updated payout policy with a game tree.

● No, but not for the reason above

No, because the game is no longer zero-sum: your opponent obtaining more coins does not necessarily make you worse off, and vice versa. We can still model this game with a game-tree, where each node contains a tuple of two values, instead of a single value. But this means the tree is no longer a minimax tree.

An example of using the minimax tree but not optimizing the number of coins collected: when given a choice between gathering 3 coins or stealing 2 coins from the opponent, the minimax solution with $c_1 - c_2$ will steal the 2 coins (net gain of 4), even though this will cause it to end up with fewer coins.

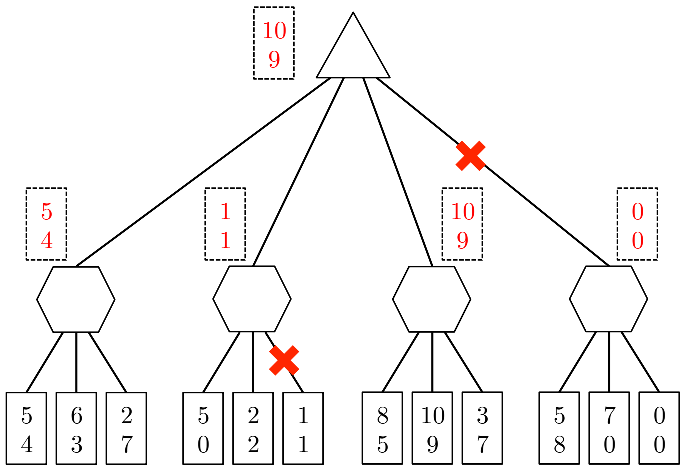
Q2. Game Trees and Pruning

You and one of the 188 robots are playing a game where you both have your own score. In the leaf nodes of the game tree, your score is on top and the robot's score on the bottom.

- The maximum possible score for either player is 10.
- You are trying to maximize your score, and you do not care what score the robot gets.
- The robot is trying to minimize the absolute difference between the two scores. In the case of a tie, the robot prefers a lower score. For example, the robot prefers (5,3) to (6,3); it prefers (5,3) to (0,3); and it prefers (3,3) to (5,5).

The figure below shows the game tree of your max node followed by the robots nodes for your four different actions.

(a) Fill in the dashed rectangles with the pair of scores preferred by each node of the game tree.



(b) You can save computation time by using pruning in your game tree search. On the game tree above, put an 'X' on line of branches that do not need to be explored. Assume that branches are explored from left to right.

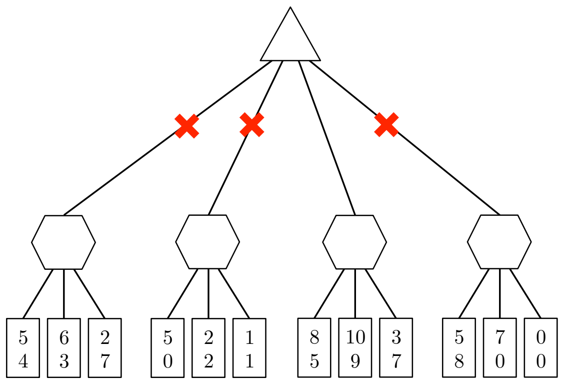
Pruning Principle: A node can be pruned if and only if its value will not affect the results at the root node, given the nodes that have already been visited.

After visiting the first subtree, the best state at the root node is $(5, 4)$. This indicates that player 1 (you) will receive at least 5 points. Any subsequent nodes explored in this game must cause the score to exceed this value to be relevant.

The next step is to visit the second subtree. After visiting $(2, 2)$, we can infer that the only better choice for the robot is (x, x), where $x < 2$. Since the maximum possible score for player 1 in this case will always be less than 5, we can prune the remaining nodes in this subtree.

After visiting the third subtree, the best state at the root node becomes $(10, 9)$. This informs us that player 1 (you) will receive at least 10 points. Given that 10 is the maximum possible score in the game, there's no need to explore further, so we prune the last subtree.

(c) You now have access to an oracle that tells you the order of branches to explore that maximizes pruning. On the copy of the game tree below, put an 'X' on line of branches that do not need to be explored given this new information from the oracle.



The given problem involves using an oracle's guidance to rearrange the exploration order of the nodes in a game tree to maximize pruning. This means we're looking for the optimal sequence that minimizes the number of nodes that need to be explored.

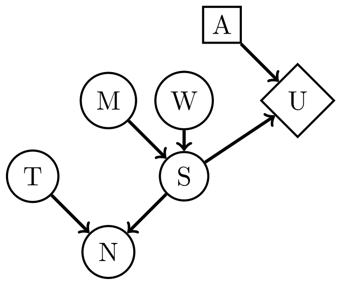
To achieve this, we can first visit the third subtree, where the best state at the root node is set to be $(10, 9)$. This corresponds to the maximum possible score in the game for player 1, thus allowing us to stop searching further.

But what about other nodes, such as $(3, 7)$? Can we prune this after visiting $(10, 9)$? The answer is no. If this node were instead $(3, 3)$, the robot would prefer $(3, 3)$ to $(10, 9)$, resulting in a different action. This scenario illustrates that we need to visit at least 3 nodes, even with the oracle's guidance.

Q3. Decision Networks and VPI

(a) This question involves d-separation of Bayes Nets. Please review related content before working on this question.

Consider the decision network structure given below:



Mark all of the following statements that **could possibly be true**, for some probability distributions for $P(M), P(W), P(T), P(S|M,W)$, and $P(N|T,S)$ and some utility function $U(S,A)$:

(i) ☐ $\text{VPI}(T) < 0$ [✓ $\text{VPI}(T) = 0$] ☐ $\text{VPI}(T) > 0$ [✓ $\text{VPI}(T) = \text{VPI}(N)$]
From the decision network, T and U are D-separated so $T \perp\!\!\!\perp U$. So $VPI(T) = 0$. VPI(N) could also be zero if N and S are independent.

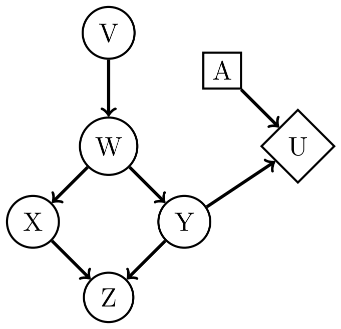
(ii) ☐ $\text{VPI}(T|N) < 0$ [✓ $\text{VPI}(T|N) = 0$] [✓ $\text{VPI}(T|N) > 0$] [✓ $\text{VPI}(T|N) = \text{VPI}(T|S)$]
From the decision network, T and U are *not* D-separated given N. So T and U can be either independent or dependent, $VPI(T|N) \geq 0$.

T and U are D-separated given S, so $T \perp\!\!\!\perp U|S$ and $VPI(T|S) = 0$. Because $VPI(T|N)$ could possibly be 0 too, it is also possible that $VPI(T|N) = VPI(T|S)$.

(iii) [✓ $\text{VPI}(M) > \text{VPI}(W)$] ☐ $\text{VPI}(M) > \text{VPI}(S)$ [✓ $\text{VPI}(M) < \text{VPI}(S)$] ☐ $\text{VPI}(M|S) > \text{VPI}(S)$ Both M and W are D-connected with U, so $VPI(M)$ and/or $VPI(W)$ could possibly be positive. So it is possible that $\text{VPI}(M) > \text{VPI}(W)$.

From the decision network, M and U are D-separated given S, meaning $M \perp\!\!\!\perp U|S$. So $VPI(M|S) = 0$. We have $VPI(M) \leq VPI(M,S) = VPI(M|S) + VPI(S) = VPI(S)$.

(b) Consider the decision network structure given below.



Mark all of the following statements that are **guaranteed to be true**, regardless of the probability distributions for any of the chance nodes and regardless of the utility function.

(i)

☐ $\text{VPI}(Y) = 0$ Observing Y could increase MEU

☐ $\text{VPI}(X) = 0$ Y can depend on X because of the path through W

☐ $\text{VPI}(Z) = \text{VPI}(W,Z)$ Consider a case where Y is independent of Z but not independent of W. Then $VPI(Z) = 0 < VPI(W,Z)$

■ VPI(Y) = VPI(Y, X) After Y is revealed, X will add no more information about Y.

(ii)

■ VPI(X) $\leq$ VPI(W) $VPI(W \mid X) + VPI(X) = VPI(X, W) = VPI(X \mid W) + VPI(W)$. We know $VPI(X \mid W) = 0$, since X is conditionally independent of Y, given W. So $VPI(W \mid X) + VPI(X) = VPI(W)$. Since VPI is non-negative, $VPI(W \mid X) \geq 0$, so $VPI(X) \leq VPI(W)$.

■ VPI(V) $\leq$ VPI(W) Since the only path from V to Y is through W, revealing V cannot give more information about Y than revealing W.

☐ VPI($V \mid W$) = VPI(V) $VPI(V \mid W) = 0$ by conditional independence, but $VPI(V)$ is not necessarily 0

☐ VPI($W \mid V$) = VPI(W) Consider a case where W is a deterministic function of V and Y is a deterministic function of W, then $VPI(W \mid V) = 0 \neq VPI(W)$

(iii)

■ VPI($X \mid W$) = 0 X is independent of Y given W

☐ VPI($Z \mid W$) = 0 Y could depend on Z, given W

■ VPI(X, W) = VPI(V, W) Both are equal to VPI(W), since both X and V are conditionally independent of Y given W.

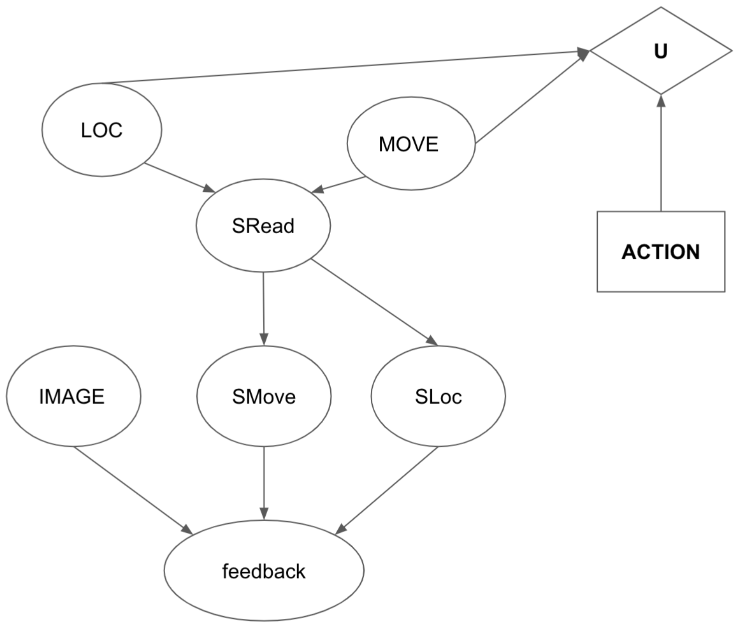
☐ VPI(W, Y) = VPI(W) + VPI(Y) VPI(W,Y) = VPI(Y), and we can have $VPI(W) > 0$

Q4. Vehicle Perception Indication

A vehicle is trying to identify the situation of the world around it using a set of sensors located around the vehicle.

Each sensor reading (SRead) is based off of an object's location (LOC) and an object's movement (MOVE). The sensor reading will then produce various values for its predicted location (SLoc) and predicted movement (SMove). The user will receive these readings, as well as the the image (IMAGE) as feedback.

(a) The vehicle takes an action, and we assign some utility to the action based on the object's location and movement. Possible actions are MOVE TOWARDS, MOVE AWAY, and STOP. Suppose the decision network faced by the vehicle is the following.



(i) Based on the diagram above, which of the following **could possibly be** true?

- [✓] VPI (Image) $= 0$
- ☐ VPI (SRead) < 0
- ☐ VPI $(\text{SMove}, \text{SRead}) >$ VPI (SRead)
- [✓] VPI (Feedback) $= 0$
- ○ None of the above

VPI(Image) $= 0$ because there is not active path connecting Image and U

VPI cannot be negative, so option 2 is not selected.

VPI(SMove, SRead) $=$ VPI(SMove $|$ SRead) $+$ VPI(SRead), therefore we can cancel VPI(SRead) from both side, and it becomes asking if VPI(SMove $|$ SRead) > 0. And we can see that cannot be true, because shading in SRead, there is no active path connecting SMove and U.

There is an active path connecting Feedback and U, therefore VPI(Feedback) ≥ 0. It could still be 0 because active path only gives the possibility of > 0.

(ii) Based on the diagram above, which of the following **must necessarily be** true?

- [✓] VPI (Image) $= 0$
- ☐ VPI (SRead) $= 0$
- [✓] VPI $(\text{SMove}, \text{SRead}) =$ VPI (SRead)
- ☐ VPI (Feedback) $= 0$

○ None of the above

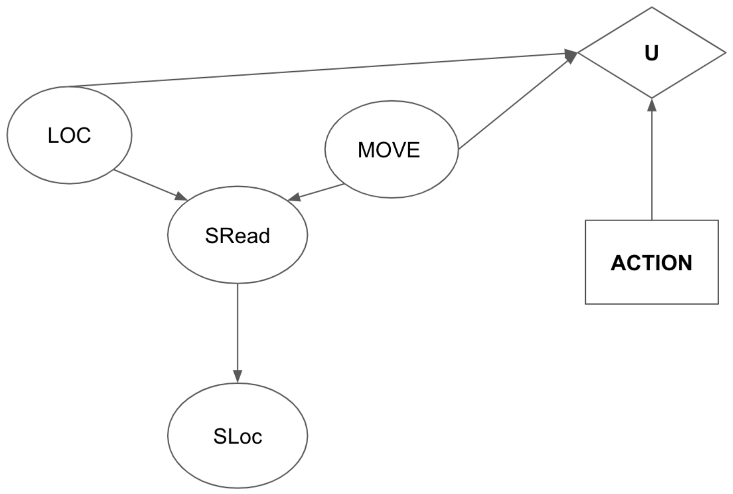
VPI(Image) = 0 because there is not active path connecting Image and U

VPI(SRead) could be > 0 because SRead-MOVE-U is an active path between SRead and U

VPI(SMove, SRead) = VPI(SMove | SRead) + VPI(SRead), therefore we can cancel VPI(SRead) from both side, and it becomes asking if VPI(SMove | SRead) == 0. And we can see that must true, because shading in SRead, there is no active path connecting SMove and U.

VPI(Feedback) could be > 0 because feedback-SLoc-SRead-MOVE-U is an active path

Let's assume that your startup has less money, so we use a simpler sensor network. One possible sensor network can be represented as follows.



You have distributions of $P(\text{LOC}), P(\text{MOVE}), P(SRead|\text{LOC}, \text{MOVE}), P(SLoc|SRead)$ and utility values $U(a, l, m)$.

(b) Complete the equation for determining the expected utility for some ACTION a.

$EU(a) = \Big(\underline{\quad\textbf{(i)}\quad}\ \underline{\quad\textbf{(ii)}\quad}\ \underline{\quad\textbf{(iii)}\quad}\Big)\ U(a, l, m)$

(i) ● $\sum_l P(l)$ ○ $\sum_{sloc} P(sloc|l)$ ○ $\sum_l \sum_{sloc} P(sloc|l)$ ○ 1

(ii) ● $\sum_m P(m)$ ○ $\sum_m P(sloc|m)$ ○ $\sum_l \sum_m \sum_{sloc} P(sloc|l)P(sloc|m)$ ○ 1

(iii) ○ $*\sum_l \sum_m \sum_{sloc} P(sloc|l)P(sloc|m)$ ○ $+\sum_l \sum_m \sum_{sloc} P(sloc|l)P(sloc|m)$
○ $+\sum_l \sum_m \sum_{sloc} P(sloc|l, m)P(l)P(m)$ ● $*1$

$EU(a) = \sum_l P(l) \sum_m P(m)U(a, l, m)$

We can eliminate SRead and SLoc via marginalization, so they don't need to be included the expression

(c) Your colleague Bob invented a new sensor to observe values of $SLoc$.

(i) Suppose that your company had no sensors till this point. Which of the following expression is equivalent to VPI($SLoc$)?

■ $\text{VPI}(SLoc) = (\sum_{sloc} P(sloc)\, \text{MEU}(SLoc = sloc)) - \max_a \text{EU}(a)$

■ $\text{VPI}(SLoc) = \text{MEU}(SLoc) - \text{MEU}(\emptyset)$

☐ $\text{VPI}(SLoc) = \max_{sloc} \text{MEU}(SLoc = sloc) - \text{MEU}(\emptyset)$

○ None of the above

Option 2 is correct by definition, and option 1 is the expanded version of option 2

(ii) Gaagle, an established company, wants to sell your startup a device that gives you $SRead$. Given that you already have Bob's device (that gives you $SLoc$), what is the maximum amount of money you should pay for Gaagle's device? Suppose you value \$1 at 1 utility.

☐ VPI($SRead$)

■ VPI($SRead$) – VPI($SLoc$)

☐ VPI($SRead, SLoc$)

■ VPI($SRead, SLoc$) – VPI($SLoc$)

○ None of the above

The answer for this question should be $VPI(\text{SRead}|\text{SLoc})$

$$\begin{aligned} VPI(\text{SRead}|\text{SLoc}) &= VPI(\text{SRead},\text{SLoc}) - VPI(\text{SLoc}) && (1) \\ &= VPI(\text{SLoc}|\text{SRead}) + VPI(\text{SRead}) - VPI(\text{SLoc}) && (2) \\ &= 0 + VPI(\text{SRead}) - VPI(\text{SLoc}) && (3) \end{aligned}$$

The first and the second equal sign are derived from the formula: $VPI(A|B) = VPI(A,B) - VPI(B)$ To show why the last equal sign is correct, i.e., $VPI(\text{SLoc}|\text{SRead})$, we look at the decision network. SLoc and U are D-separated given SRead, so SLoc $\perp\!\!\!\perp U|$SRead and $VPI(\text{SLoc}|\text{SRead}) = 0$.