CSE175 Introduction to Artificial Intelligence — Homework set #2

Fall semester 2024 — Miguel Á. Carreira-Perpiñán

This set covers chapters 5–6 of the AIMA textbook (4th ed.). References (figures, etc.) are to the AIMA textbook. In all exercises, *explain your answers*. **Total possible points: 100.**

Exercise 1 (30 points). Consider the following variation of tic-tac-toe. The board has 1×4 cells and there are two players that alternate placing one piece at a time on the board. One player has pieces of the form $\times$ and the other of the form $\circ$. Whenever a player has two consecutive pieces on the board it wins; if the board fills without consecutive pieces, the game is a tie. Player $\times$ starts. Assume terminal game values of $-1/+1/0$ for loss/win/tie for the starting player. An example game: $\begin{array}{|c|c|c|c|}\hline \ & \times & \ & \ \\ \hline\end{array} \rightarrow \begin{array}{|c|c|c|c|}\hline \ & \times & \circ & \ \\ \hline\end{array} \rightarrow \begin{array}{|c|c|c|c|}\hline \times & \times & \circ & \ \\ \hline\end{array} \rightarrow$ player $\times$ wins.

1. (10 points) Solve this game by constructing its complete minimax tree. Draw the tree as in fig. 5.1. Give the optimal value v^* at the root, and the corresponding move(s).
2. (2 points) Explain how a real game would develop, i.e., give the move sequence that would happen according to your tree.

Assume now that player $\times$ plays randomly in the following way: in its first move, it places a piece in one of cells 1, 2, 3, 4 with probability q, p, p, q, respectively, for some fixed values $p, q \in [0, 1]$; in any subsequent moves, it picks uniformly at random among all possible moves. Player $\circ$ plays optimally as usual (to minimize).

3. (11 points) Solve this game by constructing its complete minimax tree. Draw the tree as in fig. 5.1. Give the optimal value $v^*(p)$ (as a function only of p) at the root, and the corresponding move(s).
4. (5 points) Explain how a real game would develop; give all the possible games (move sequences) that can happen according to your tree and the probability of each game.
5. (2 points) Since $v^*(p)$ depends on p, find the maximum and minimum value that v^* can achieve and for what value of p.

Exercise 2 (20 points). Consider the function

$$f(x, y, z) = \max(\min(7, 10, 9), \min(x, 10, y, 6, 12, 1), \min(\max(11, 3), z)) \qquad x, y, z \in \mathbb{R}.$$

1. (5 points) Using mathematical properties, simplify f and plot it (i.e., evaluate it at all its argument values and show it graphically). Hint: it really depends on one variable only.
2. (2 points) Draw the function as a minimax tree as in fig. 5.2.

Set $x = 8$, $y = 2$ and $z = 9$.

3. (7 points) Compute $f(x, y, z)$ by using the minimax search with alpha-beta pruning. Show the values of the bounds α and β as the algorithm progresses. How many nodes are pruned (not evaluated)? Indicate them in the tree.
4. (3 points) Draw an equivalent tree (i.e., representing f exactly) by reordering children such that alpha-beta pruning will prune as many nodes as possible. How many nodes are pruned? Indicate them in the tree.
5. (3 points) Same but such that alpha-beta pruning will prune as few nodes as possible.

Exercise 3 (10 points). Consider a CSP over variables $X_1, \dots, X_8$, each over the domain $\{0,1,2,3,4,5\}$ and with the following binary constraints:

$$X_1 < X_2 \qquad X_5 \neq X_6 \qquad X_2 < X_6 \qquad X_6 = X_7 - 1 \qquad X_7 = X_4 \qquad X_7 \leq X_8 - 3 \qquad X_7 + X_3 \leq 3.$$

1. (1 points) Draw the constraint graph. Is it a tree?
2. (9 points) Solve the CSP (i.e., find a solution if one exists) by using the tree CSP algorithm (fig. 6.11). Be explicit about every step in the algorithm (indicate how you construct the topological sort, how the domain of each variable evolves, how you pick a value for each variable, etc.). If needed to break ties within any step, use the order of the variables.

Exercise 4 (40 points). Consider as CSP the 2-SAT problem, which consists of finding an assignment of binary variables (if one exists) that satisfies a CNF sentence (conjunctive normal form) with clauses involving two variables (i.e., binary constraints). Specifically, consider the CNF sentence over the binary variables $p_1, \dots, p_6 \in \{0,1\}$:

$$(\underbrace{p_1 \vee p_2}_{\text{A}}) \wedge (\underbrace{p_1 \vee p_3}_{\text{B}}) \wedge (\underbrace{p_2 \vee \neg p_4}_{\text{C}}) \wedge (\underbrace{\neg p_1 \vee p_4}_{\text{D}}) \wedge (\underbrace{p_5 \vee \neg p_6}_{\text{E}}).$$

In the questions below, some of the algorithms need to pick or scan variables, values or constraints. Unless otherwise stated, pick them in their natural order (for variables: first p_1, then p_2, etc.; for values, first 0 then 1; for constraints: first A, then B, etc.).

1. (2 points) Draw the constraint graph.
2. (5 points) Apply arc consistency to the graph (algorithm AC-3). Comment on the result.

Solve this problem in each of the following ways:

3. (5 points) By brute force, i.e., enumerate all possible assignments and find which ones satisfy all constraints. How many possible assignments are there, and how many solutions?
4. (8 points) By backtracking (section 6.3), using all the heuristics in sections 6.3.1 (minimum-remaining-values then degree heuristic to choose variables, least-constraining-value to choose values) and 6.3.2 (forward checking). Draw the search tree.
5. (8 points) By local search (section 6.4), starting with all variables assigned to 0, scanning variables in order; for each variable, set it to the value (0 or 1) that minimizes the number of conflicts (if 0 and 1 are tied, pick 0). Repeat by scanning variables in reverse order (first p_6, then p_5, etc.). Explain what happens.
6. (12 points) By cutset conditioning (section 6.5.1). To create the cutset, scan the variables in order.