1 Trains

A train scheduler must decide when trains A, B and C should depart. Once a train departs, it moves one space along its track each hour (in discrete jumps) until it arrives at its destination platform. Each train can depart at 1, 2 or 3 pm. The scheduler has two restrictions: All trains must leave at different times, and two trains should not both occupy crossing sections of track after any one hour time step is over. Note that train A is two spaces long. Also note that the collision constraint is enforced only at the conclusion of every hour - time is discrete in this problem.

(a) Describe the constraint satisfaction problem that, when solved, will tell the train scheduler when each train should depart. Let the variables A, B and C represent the departure times of the three trains.

(b) Draw the constraint graph for the CSP you defined.

(c) After selecting A = 2, cross out all values for B and C eliminated by forward checking.

\[
\begin{array}{ccc}
A & B & C \\
2 & 1 & 2 & 3 & 1 & 2 & 3 \\
\end{array}
\]
(d) Cross out all values eliminated by arc consistency before assigning any variables.

\[
\begin{array}{ccc}
A & B & C \\
1 & 2 & 3 \\
1 & 2 & 3 \\
1 & 2 & 3 \\
\end{array}
\]

(e) After selecting \( A = 2 \), cross out all values for \( B \) and \( C \) eliminated by arc consistency.

\[
\begin{array}{ccc}
A & B & C \\
2 & 1 & 2 \\
2 & 1 & 3 \\
2 & 1 & 3 \\
\end{array}
\]

(f) Describe the execution of backtracking search using forward checking and the minimum remaining values (MRV) and least constraining values (LCV) heuristics. Specifically, in what order are the variables assigned and what values do they take? Start by assigning variable \( A \). You may not need to fill all the lines below:

(1) variable \( A \) is assigned value .
(2) variable  is assigned value .
(3) variable  is assigned value .
(4) variable  is assigned value .
(5) variable  is assigned value .
(6) variable  is assigned value .
2 Minimax Search

In this problem, we will explore adversarial search.

Consider the zero-sum game tree shown below. Trapezoids that point up, such as at the root, represent choices for the player seeking to maximize; trapezoids that point down represent choices for the minimizer. Outcome values for the maximizing player are listed for each leaf node. It is your move, and you seek to maximize the expected value of the game.

(a) Assuming both opponents act optimally, carry out the minimax search algorithm. Write the value of each node inside the corresponding trapzoid. What move should you make now? How much is the game worth to you?

(b) Now reconsider the same game tree, but use $\alpha$-$\beta$ pruning (the tree is printed on the next page). Expand successors from left to right. In the brackets $[$, $]$, record the $[\alpha, \beta]$ pair that is passed down that edge (through a call to MIN-VALUE or MAX-VALUE). In the parentheses $( )$, record the value ($v$) that is passed up the edge (the value returned by MIN-VALUE or MAX-VALUE). Circle all leaf nodes that are visited. Put an ‘X’ through edges that are pruned off. How much is the game worth according to $\alpha$-$\beta$ pruning?