CS410: Artificial Intelligence 2021 Fall
Homework 3: Local Search & Adversarial Search & MDP
Due date: 23:59:59 (GMT +08:00), November 6 2021

1. **Local Search**. The traveling salesman problem (TSP) is the problem of finding the shortest route to visit a set of cities exactly once and return to the starting city. Describe how to use genetic algorithm for TSP. Propose a state representation, the corresponding crossover and mutation, and the fitness function.

   Solution: Suppose there are n cities.

   - State Representation: We denote each city by a unique number in $\{1, 2, \ldots, n\}$ and we represent one route by an ordered sequence $S^k = \{s_i^k\}_{i=1}^n$ which is a permutation over the set $\{1, 2, \ldots, n\}$. Furthermore, $s_i^k \in \{1, 2, \ldots, n\}$ indicates that s_i^k is the i-th city to be visited in route S^k.
   - Crossover: First randomly generate a cut point $p \in \{1, 2, \ldots, n-1\}$ and exchange the top p elements in S^{k_1} and S^{k_2} to get 2 new sequences $S^{l_1} = S^{k_1}_{[1:p]} + S^{k_2}_{[p+1:n]}$ and $S^{l_2} = S^{k_2}_{[1:p]} + S^{k_1}_{[p+1:n]}$, where $+$ here denotes the concatenation operation between 2 sequences and $S_{[i:j]}$ indicates slicing the sequence S from index i to index j. Then eliminate the repeated elements in S^{l_1} and S^{l_2} simultaneously by exchanging the repeated elements. For instance, if there are two elements in S^{l_1} with the same value (say a), then there must also be at least one pair of elements in S^{l_2} with the same values (say b). Then we swap one element in S^{l_1} with value a and one element in S^{l_2} with value b. Repeat until there is no repetition.
   - Mutation: For some sequence S^k, randomly exchange the values of 2 elements in it with a (usually small) independent probability.
   - Fitness Function: One choice of the fitness functions could be $f(S^k) = \frac{1}{\text{len}(S^k)}$, where $\text{len}(S^k)$ denotes the sum of distances between visited cities in this sequence(route). If there are any 2 cities disconnected from each other in this sequence, let $\text{len}(S^k) = \infty$.

   Reasonable answers are acceptable.

2. **Game Tree**. Prove that with a positive linear transformation of leaf values (i.e., transforming a value x to $ax + b$ where $a > 0$), the choice

of move remains unchanged in a game tree, even when there are chance nodes.

Solution: The general strategy is to reduce a general game tree to a one-ply tree by induction on the depth of the tree. The inductive step must be done for min, max, and chance nodes, and simply involves showing that the transformation is carried though the node. Suppose that the values of the descendants of a node are $x_1, \ldots, x_n$, and that the transformation is $ax + b$, where a is positive. We have

$$\begin{aligned}
&\min(ax_1 + b, ax_2 + b, \ldots, ax_n + b) = a \min(x_1, x_2, \ldots, x_n) + b \\
&\max(ax_1 + b, ax_2 + b, \ldots, ax_n + b) = a \min(x_1, x_2, \ldots, x_n) + b \\
&p_1(ax_1 + b) + p_2(ax_2 + b) + \cdots + p_n(ax_n + b) \\
&= a(p_1x_1 + p_2x_2 + \cdots p_nx_n) + b
\end{aligned}$$

Hence the problem reduces to a one-ply tree where the leaves have the values from the original tree multiplied by the linear transformation. Since $x > y \Rightarrow ax + b > ay + b$ if $a > 0$, the best choice at the root will be the same as the best choice in the original tree.

Reasonable answers are acceptable.

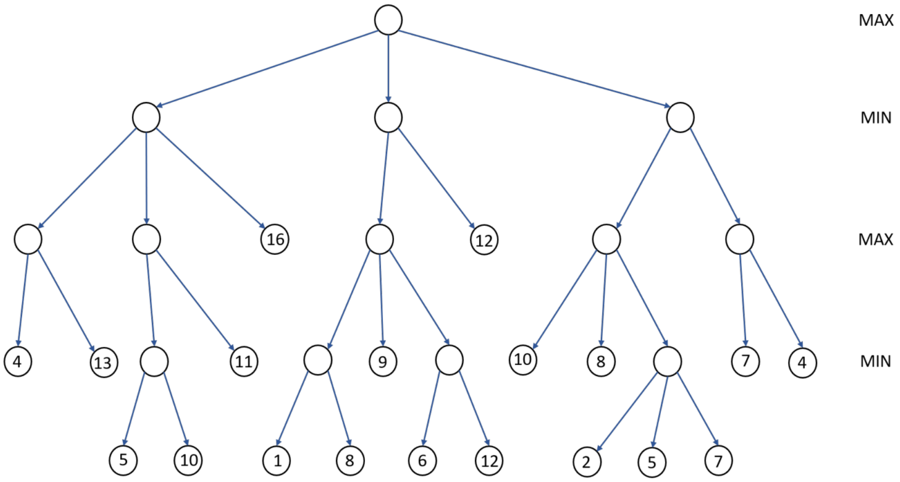

Figure 1: Problem 3.

3. **Alpha-Beta Pruning**. Consider the above game tree.

   (a) Compute the minimax value for each node using Minimax algorithm.

   (b) Prune the game tree using Alpha-Beta pruning algorithm. Provide the final alpha and beta values computed at the root, each internal node visited, and at the top of pruned branches. Provide the pruned branches. Assume child nodes are visited from left to right.

Solution:

(a) Please refer to Figure 2.

(b) Please refer to Figure 3.

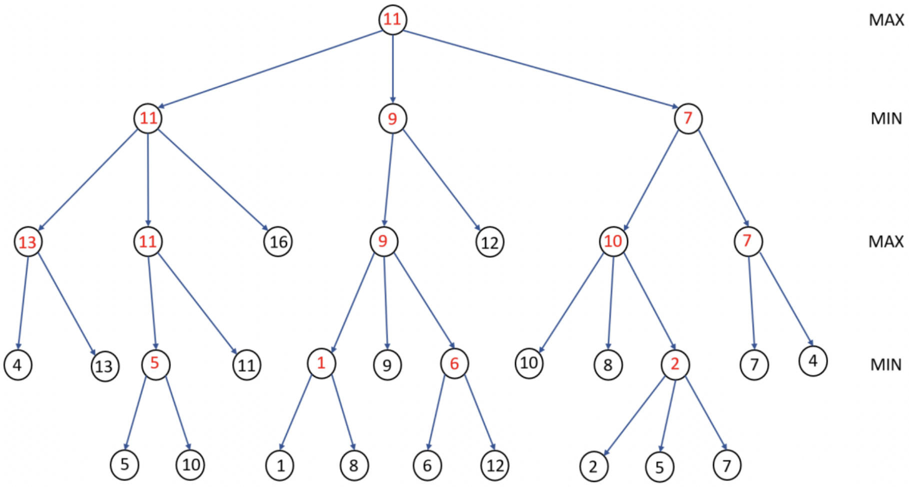

Figure 2: Solution for Problem 3 (a).

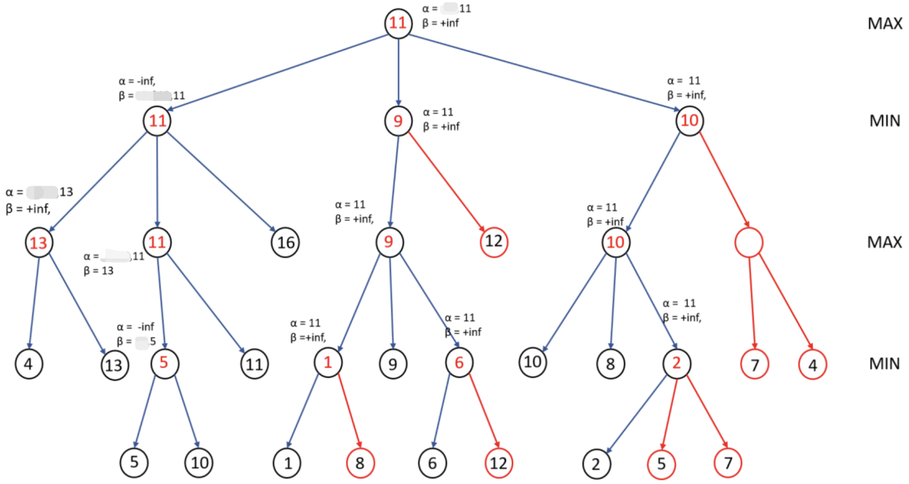

Figure 3: Solution for Problem 3 (b), where the red branches are the pruned branches.

4. **Racing Problem**. Consider the racing problem in Page 17, Lecture 6.

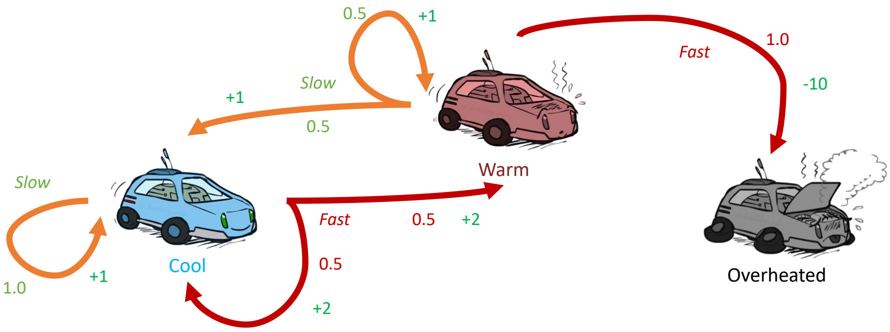

Figure 4: Problem 4.

Assume there is a discount factor $0 < \gamma < 1$ in the MDP of this problem. Calculate $V^*(s)$ for each state s and $Q^*(s, a)$ for each q-state (s, a) in this problem.

Solution: Denote by s_1, s_2 and s_3 the states Cool, Warm and Overheated respectively. Let a_1, a_2 be the action Fast and Slow respectively. It is clear that the optimal policy π^* satisfies $\pi^*(s_1) \to a_1$ and $\pi^*(s_2) \to a_2$. By the definition of V^*, one can see that

$$
\begin{aligned}
&V^*(s_1) = 0.5(\gamma V^*(s_2) + 2) + 0.5(\gamma V^*(s_1) + 2)\\
&V^*(s_2) = 0.5(\gamma V^*(s_2) + 1) + 0.5(\gamma V^*(s_1) + 1)\\
&V^*(s_3) = 0\,.
\end{aligned}
$$

Solving this system of linear equations shows that $V^*(s_1) = \frac{4-\gamma}{2-2\gamma}$, $V^*(s_2) = \frac{2+\gamma}{2-2\gamma}$, $V^*(s_3) = 0$. Furthermore, the definition of Q^* implies that $Q^*(s_1, a_1) = V^*(s_1) = \frac{4-\gamma}{2-2\gamma}$, $Q^*(s_1, a_2) = 1 + \gamma V^*(s_1) = \frac{2+2\gamma-\gamma^2}{2-2\gamma}$, $Q^*(s_2, a_1) = -10$ and $Q^*(s_2, a_2) = V^*(s_2) = \frac{2+\gamma}{2-2\gamma}$.

5. **Convergence of Policy Iteration**. Assume that the environment is a finite MDP (i.e., its state, action, and reward sets are finite). Prove that each policy improvement must produce a new policy as good as, or better than the original one. Prove policy iteration converges to an optimal policy.

Solution: 1. Prove that each policy improvement must produce a new policy as good as, or better than the original one:

Theorem 1 (Policy Improvement Theorem)**.** *Let π and π' be any pair of policies such that, for all $s \in \mathcal{S}$,*

$$
Q_\pi\left(s, \pi'(s)\right) \geq V_\pi(s)\,. \tag{1}
$$

Then the policy π' must be as good as, or better than, π. That is, it must obtain greater or equal expected return from all states $s \in \mathcal{S}$:

$$V_{\pi'}(s) \geq V_{\pi}(s)\,. \tag{2}$$

Moreover, if there is strict inequality of Eq. (1) *at any state, then there must be strict inequality of Eq.* (2) *at that state.*

Proof of Theorem 1.

$$\begin{aligned}
V^{\pi}(s) &\leq Q^{\pi}\left(s, \pi'(s)\right) \\
&= E_{\pi'}\left\{r_{t+1} + \gamma V^{\pi}\left(s_{t+1}\right) \mid s_t = s\right\} \\
&\leq E_{\pi'}\left\{r_{t+1} + \gamma Q^{\pi}\left(s_{t+1}, \pi'\left(s_{t+1}\right)\right) \mid s_t = s\right\} \\
&= E_{\pi'}\left\{r_{t+1} + \gamma E_{\pi'}\left\{r_{t+2} + \gamma V^{\pi}\left(s_{t+2}\right)\right\} \mid s_t = s\right\} \\
&= E_{\pi'}\left\{r_{t+1} + \gamma r_{t+2} + \gamma^2 V^{\pi}\left(s_{t+2}\right) \mid s_t = s\right\} \\
&\leq E_{\pi'}\left\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 V^{\pi}\left(s_{t+3}\right) \mid s_t = s\right\} \\
&\vdots \\
&\leq E_{\pi'}\left\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} + \cdots \mid s_t = s\right\} \\
&= V^{\pi'}(s).
\end{aligned}$$

□

Recall that in the policy improvement step we greedily construct the new policy π' such that

$$\pi'(s) \doteq \arg\max_a Q_{\pi}(s,a)\,,$$

which satisfies that

$$\begin{aligned}
Q_{\pi}(s, \pi'(s)) &= \max_a Q_{\pi}(s,a) \\
&\geq Q_{\pi}(s, \pi(s)) \\
&= V_{\pi}(s)\,.
\end{aligned}$$

Hence π' meets the conditions of Theorem 1 and applying Theorem 1 concludes the proof.

2. Prove policy iteration converges to an optimal policy:

The proof consists of two parts. We first show that policy iteration will converge to some policy and then prove that the convergent policy is exactly the optimal policy π^*.

Because a finite MDP has only a finite number of policies and the policy improvement produces a new policy as good as, or better than the policy in last iteration, policy iteration must converge to some policy $\bar{\pi}$ in a finite number of iterations.

Now suppose the convergent policy $\bar{\pi}$, is as good as, but not better than, the old policy π. Then it holds that

$$\begin{aligned} V_{\bar{\pi}}(s) &= \max_a \mathbb{E}\left[r_t + \gamma V_{\bar{\pi}}\left(S_t\right) \mid S_t = s, A_t = a\right] \\ &= \max_a \sum_{s', r} p\left(s', r \mid s, a\right)\left[r + \gamma V_{\bar{\pi}}\left(s'\right)\right] \ . \end{aligned}$$

But this is the same as the Bellman optimality equation, and therefore, $V_{\bar{\pi}}$ must be V^*, and both π and $\bar{\pi}$ must be optimal policies.

Reasonable answers are acceptable.