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:

   (a) **State:**$S = \{s_1, s_2, s_3 \dots s_n\}$ is a rotation of the $[n]$ and represents the route. The salesman travels the city s_i at the i^{th} time and returns to city s_1.

   (b) **Crossover:**Given two state S^1, S^2, we run the crossover algorithm to get two new states $\overline{S}^1, \overline{S}^2$.

      i. Assume $S^1 = \{s_1^1, \dots s_n^1\}, S^2 = \{s_1^2, \dots s_n^2\}$, we generate two new temporary states
      $S^{1'} = \{s_1^1, \dots s_{\lfloor \frac{n}{2} \rfloor}^1, s_{\lfloor \frac{n}{2} \rfloor + 1}^2, \dots s_n^2\}$
      $S^{2'} = \{s_1^2, \dots s_{\lfloor \frac{n}{2} \rfloor}^2, s_{\lfloor \frac{n}{2} \rfloor + 1}^1, \dots s_n^1\}$.

      ii. There must be repetitive city in $S^{1'}$. Assume $\exists p <= \lfloor \frac{n}{2} \rfloor, q >= \lfloor \frac{n}{2} \rfloor + 1$ satisfying $S^{1'}[p] = S^{1'}[q]$, modify the $S^{1'}$ that $S^{1'}[p] = S^2[p]$. Repeat this operation until there is no repetitive elements in $S^{1'}$, then we get the $\overline{S}^1$.

      iii. Also applying the same methods on $S^{2'}$ and get the $\overline{S}^2$.

   (c) **Mutation:** For a given state S, we randomly choose 2 elements $1 \leq i < j \leq n$ that exchange the s_i, s_j in S and get the new state.

   (d) **Fitness Function:**Let $S = \{s_1, s_2, s_3 \dots s_n\}$, and we define the fitness function f that

   $$f(S) = \frac{1}{\sum_{i=1}^{n} d(s_i, s_{i+1})} \quad with \quad s_{n+1} := s_1 \tag{1}$$

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

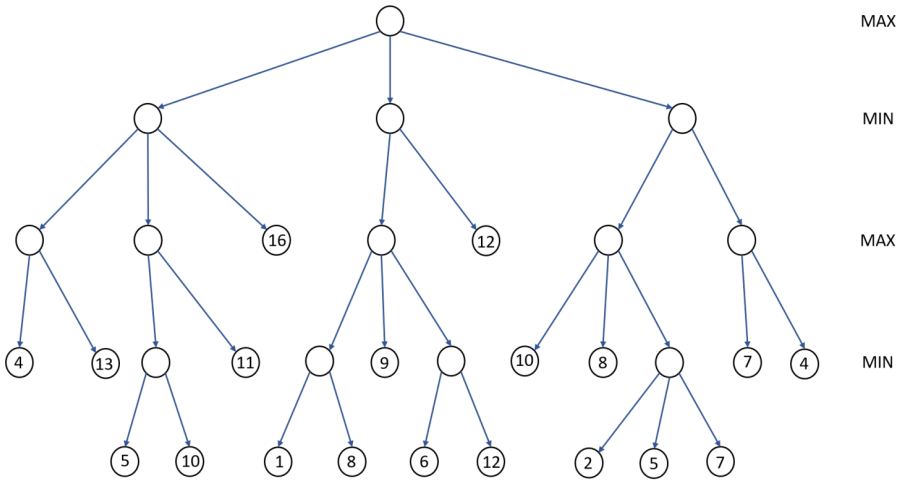

Figure 1: Problem 3.

nodes.

Solution:
Let f be the positive linear transformation that $f(x) = a*x+b$. We prove it step by step.

(a) $x_1 \leq x_2 \Leftrightarrow f(x_1) \leq f(x_2)$.That's because

$$x_1 \leq x_2 \Leftrightarrow a * x_1 \leq a * x_2 \Leftrightarrow f(x_1) \leq f(x_2) \quad with \quad a > 0 \tag{2}$$

(b) From (a) we know

$$\begin{aligned} \arg\max_x x &= \arg\max_x f(x) \\ \arg\min_x x &= \arg\min_x f(x) \end{aligned} \tag{3}$$

So for all the max and min points x,

$$\begin{aligned} value(x) &= \arg\max_{y \in children(x)} = \arg\max_{y \in children(x)} f(y) = f(x) \\ value(x) &= \arg\min_{y \in children(x)} = \arg\min_{y \in children(x)} f(y) = f(x) \end{aligned} \tag{4}$$

(c) For any probability sequenses $\{\alpha_1 \dots \alpha_n\}$ with $\sum_{i=1}^{n} \alpha_i = 1$, and a value

sequense $X = \{x_1 \dots x_n\}$. We have

$$
\begin{aligned}
& f(\sum_{i=1}^{n} \alpha_i * x_i) \\
=& a * \sum_{i=1}^{n} \alpha_i * *x_i + b \\
=& \sum_{i=1}^{n} \alpha_i * (a * x_i) + b * \sum_{i=1}^{n} \alpha_i \\
=& \sum_{i=1}^{n} \alpha_i * [(a * x_i) + b] \\
=& \sum_{i=1}^{n} \alpha_i * f(x_i)
\end{aligned} \tag{5}
$$

(d) So for the chance nodes x, we also have $value(x) = f(x)$.

From (a) to (d) we can conclude that the choice of move remains unchanged.

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:
Please refer to the following figures 2

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

   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:

   (a) $V(Overheated) = 0$.That's because in every iteration, $V(Overheated) = 0 + \gamma * V(Overheated), \Rightarrow V(Overheated) = 0$.

   (b) $V^*(Warm) > 0, V^*(Slow) > 0$.That's because these two state has at least an action that lets it not transform to state **Overheated**. And these action has reward >0.

   (c) $\pi^*(Warm, Slow) = 1$.That's because if $\pi(Warm, Fast) = 1$, $V^*(Warm, Fast) = -10 < 0$.

(d) If $\pi^*(Cool, Slow) = 1$, we have

$$\begin{aligned} &V^*(Cool) = 1 + \gamma * V^*(Cool) \\ &V^*(Warm) = 0.5 * (1 + \gamma * V^*(Warm)) + 0.5 * (1 + \gamma * V^*(Cool)) \\ \Rightarrow &V^*(Cool) = V^*(Warm) = \frac{1}{1-\gamma} \end{aligned} \tag{6}$$

However, we have

$$\begin{aligned} Q^*(Cool, Fast) &= 0.5 * (2 + \gamma * V^*(Cool)) + 0.5 * (2 + \gamma * V^*(Warm)) \\ &= 1 + \frac{1}{1-\gamma} > V^*(Cool) \end{aligned} \tag{7}$$

So we can only conclude $\pi^*(Cool, Warm) = 1$,and get

$$\begin{aligned} &V^*(Cool) = 0.5 * (2 + \gamma * V^*(Warm)) + 0.5 * (2 + \gamma * V^*(Cool)) \\ &V^*(Warm) = 0.5 * (1 + \gamma * V^*(Warm)) + 0.5 * (1 + \gamma * V^*(Cool)) \\ \Rightarrow &V^*(Warm) = \frac{2+\gamma}{2-2\gamma}, V^*(Cool) = \frac{4-\gamma}{2-2\gamma} \\ \Rightarrow &Q(Cool, Fast) = \frac{4-\gamma}{2-2\gamma}, Q(Cool, Fast) = \frac{2+2\gamma-\gamma^2}{2-2\gamma} \\ \Rightarrow &Q(Warm, Fast) = -10, Q(Warm, Fast) = \frac{2+\gamma}{2-2\gamma} \end{aligned} \tag{8}$$

5. **Convergence of Policy Iteration**. Prove the policy improvement method can indeed improve policies and then prove the convergence of policy iteration.

Solution:

Assume the MDP is a finite MDP. And we use V^π to represent the value function vector of n states under the policy π. In the i^{th} iteration, we get policy π_i.

(a) We first prove if $\pi_i \neq \pi^*, V^{\pi_{i+1}} > V^{\pi_i}$.Here the $>$ means,$\forall 0 < j \leq n, V^{\pi_{i+1}}(j) \geq V^{\pi_i}(j)$ and $\exists 0 < j \leq n, V^{\pi_{i+1}}(j) > V^{\pi_i}(j)$.

Since the π_i is not optimal, the $\exists\ s$ that $V^{\pi_i}(s) = Q^{\pi_i}(s, \pi_i(s)) < Q^{\pi_i}(s, a')$. So $\pi_{i+1} \neq \pi_i$. And let $S^i = s_1^i, s_2^i \dots s_k^i$ satisfies $\forall s \in S^i$,$\pi_{i+1}(s) \neq \pi_i(s)$. Then we construct $k+1$ policies $\pi'_0, \pi'_1 \dots \pi'_k$that $\pi'_0 = \pi_i, \pi'_k = \pi_{i+1}$ and

$$\pi'_j(s) = \begin{cases} \pi'_{j-1}(s) & if \quad s = s_j^i \\ \pi_{i+1}(s) & if \quad s \neq s_j^i \end{cases} \quad with \quad 1 \leq j \leq k \tag{9}$$

Then we prove $\forall j,\ V^{\pi'_{j+1}} > V^{\pi'_j}$. Here we apply the algorithm that we initial the vector $V_0 = V^{\pi_i}$ and do the iteration $V_{t+1} = R^{\pi_i} + \gamma * P^{\pi_i} * V_t$. For this algorithm, we can get

i. Let V' satisfies $V' = R^{\pi_j} + \gamma * P^{\pi_j} * V'$. Then $V' = R^{\pi_j}/(1 - \gamma * P^{\pi_j})$ is unique and $\lim_{t\to+\infty} V_t = V'$. Here we take the ∞-norm as measure, which means $|V| = \max_s |V(s)|$. Then we have

$$\begin{aligned} V' &= R^{\pi_j} + \gamma * P^{\pi_j} * V' \\ V_{t+1} &= R^{\pi_j} + \gamma * P^{\pi_j} * V_t \\ \Rightarrow V_{t+1} - V' &= \gamma * P^{\pi_j} * (V_t - V') \end{aligned} \tag{10}$$

Since the every row of P^{π_i} with the sum 1. So we have

$$\begin{aligned} &|P^{\pi_j} * (V_t - V')| \leq |(V_t - V')| \\ \Rightarrow &|V_{t+1} - V'| \leq \gamma * |(V_t - V')| \end{aligned} \tag{11}$$

Then we have $\lim_{t\to+\infty} |V_t - V'| = 0$, which means

$$\lim_{t\to+\infty} V_t = V' = V^{\pi'_{j+1}} \tag{12}$$

ii. $\forall t \geq 1, V_t > V_0 = \pi'_j$. We prove this using induction. When $t = 1$, we have

$$V_1(s) = \begin{cases} Q(s, \pi_{i+1}(s)) > Q(s, \pi_i(s)) = V_0(s) & if \quad s = s^i_j \\ V_0(s) & if \quad s \neq s^i_j \end{cases} \tag{13}$$

. When $t > 1$, since $P^{\pi'_j}$ all positive, and $V_{t-1} > V_0$. So $\gamma * P^{\pi'_j} V_{t-1} > \gamma * P^{\pi'_j} V_0$, then $V_t > V_1 > V_0$.

From above proof we get

$$V^{\pi'_{j+1}} = \lim_{t\to\infty} V_t > V^{\pi'_j} \tag{14}$$

So

$$V^{\pi_{i+1}} = V^{\pi'_k} > V^{\pi'_{k-1}} \ldots V^{\pi'_0} = V^{\pi_i} \tag{15}$$

(b) $\forall \pi, \forall s,\ V^\pi(s) = \sum_{i=0}^{\infty} R_i * \gamma^i < \frac{R_{max}}{1-\gamma}$ is bounded.

(c) From the above two proof, we know if the policy $\pi_i \neq \pi^*$, then the π will change. After at most $|\mathbb{S}|^|\mathbb{A}|$, it must stop at the π^*.

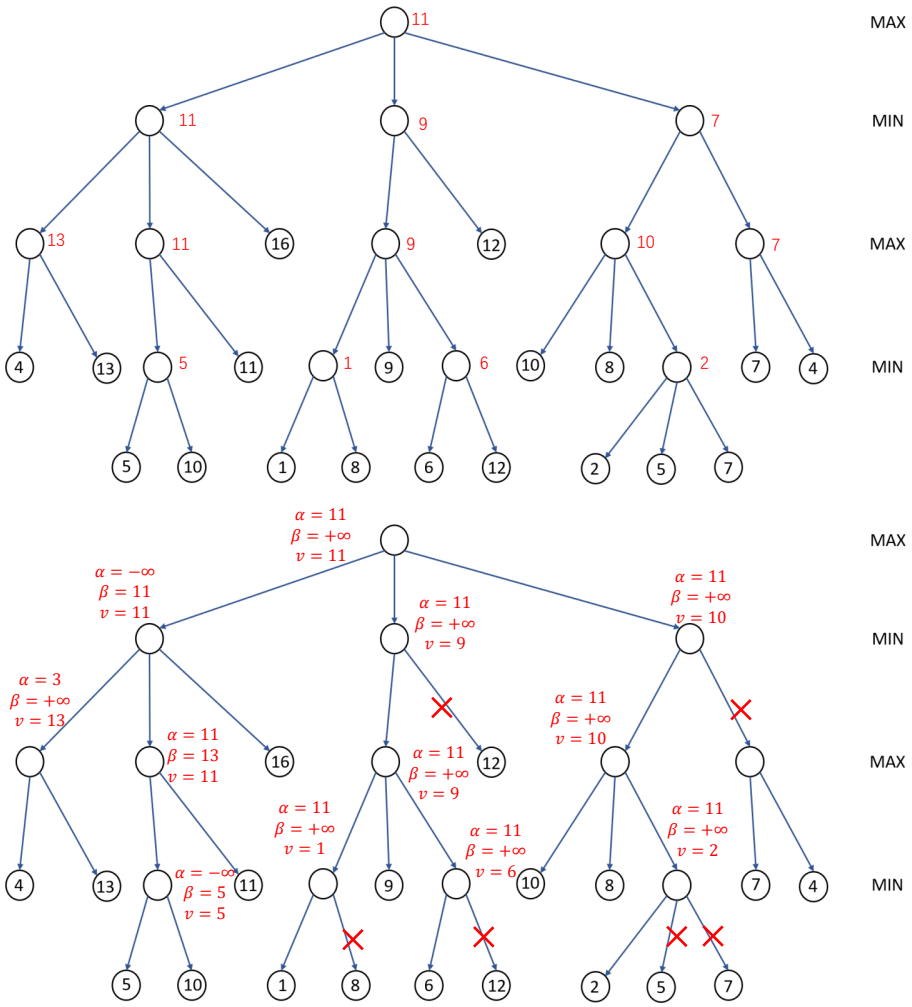

Figure 2: Solution 3

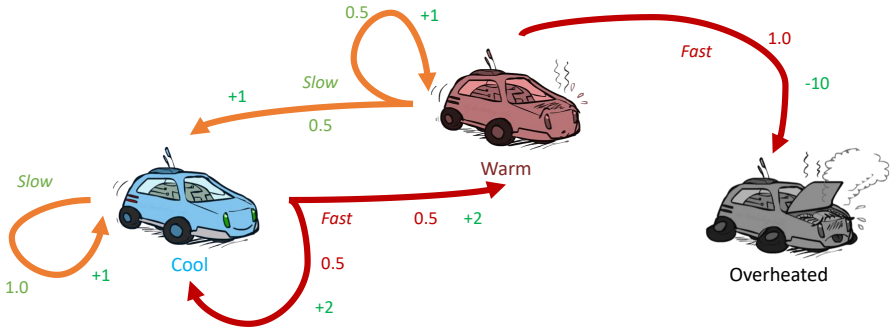

Figure 3: Problem 4.