AI Planning for Autonomy

Sample Solutions for Problem Set VIII: Monte-Carlo Tree Search

1. MCTS tree updates[1], and the tree is included at the end of this document for each step:

   I1: Select $(2,1)$; Expand N; Do simulation on $\xrightarrow{succ}(2,2)$; Backup the reward -1.

   I2: Select $(2,1)\rightarrow(2,2)$; Expand E; Do simulation on $\xrightarrow{slip(S)}(1,2)$; Backup the reward -1.

   I3: Select $(2,1)$; Expand E; Do simulation on $\xrightarrow{succ}(3,1)$; Backup the reward -1.

   I4: Select $(2,1)$; Expand W; Do simulation on $\xrightarrow{succ}(2,1)$; Backup the reward -1.

   I5: Select $(2,1)\rightarrow(2,2)$; Expand S; Do simulation on $\xrightarrow{slip(W)}(2,2)$; Backup the reward -1.

2. Calculate using $\text{argmax}_{a\in A}Q(s,a)$. The answer would be W (West) because it has the highest Q-value.

   W: $Q((2,1),W)=0.8$

   E: $Q((2,1),E)=-0.8$

   N: $Q((2,1),N)=0.08$

   S: $Q((2,1),S)=0$

3. Need to calculate π for each of N, S, E, W based on the UCT formula and then normalise. However, in MCTS, normally we expand all the successors once before we run UCT forumla.

$$\pi(s) \;=\; argmax_{a\in A(s)}\begin{pmatrix} W & : & 0.8+\sqrt{\frac{2\ln 5}{1}} \\ E & : & -0.8+\sqrt{\frac{2\ln 5}{1}} \\ S & : & \infty \\ N & : & 0.08+\sqrt{\frac{2\ln 5}{3}} \end{pmatrix}$$

   Therefore, UCT would be more likely to choose S.

4. For 3-step SARSA is calculated as:

   $Q(s,a)=Q(s,a)+\alpha[G_t^n-Q(s,a)]$

   $G_t^n=r_t+\gamma\cdot t_{t+1}+\gamma^2\cdot r_{t+2}+\cdots+\gamma^n\cdot Q(S_{t+n},\pi(S_{t+n}))$

$$\begin{aligned} Q(S,P) &= Q(S,P)+0.4\cdot[G_S^3-Q(S,P)] \\ &= -0.7+0.4\cdot[-1.4716-(-0.7)] \\ &= -0.7+0.4\times(-0.7716) \\ &= -1.00864 \end{aligned}$$

   where

[1] The iteration traces in this workshop are generated by vanilla version MCTS, which are slightly different from the Lecture Slides (Select phase does not consider a not-fully expanded node as most urgent node to select in the vanilla version MCTS). However, if you choose any policy, such as UCB as Tree Policy (for selection and expansion), then based on that policy, the selection would select a not fully expanded node first.

$$
\begin{aligned}
G_S^3 &= r(S,P) + 0.9 \cdot r(M,S) + 0.9^2 \cdot r(Scored,R) + 0.9^3 \cdot Q(M,P) \\
&= (-1) + 0.9 \cdot (-2) + 0.9^2 \cdot 2 + 0.9^3 \cdot (-0.4) \\
&= (-1) + (-1.8) + 1.62 + (-0.2916) \\
&= -1.4716
\end{aligned}
$$

And basically yes, it can converge much faster than 1-step.

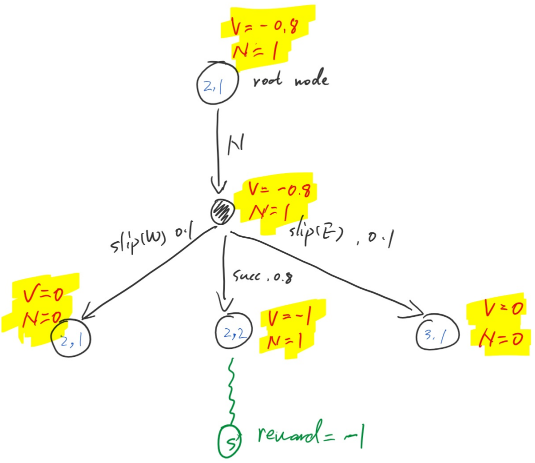

Figure 1: MCT after iteration 1

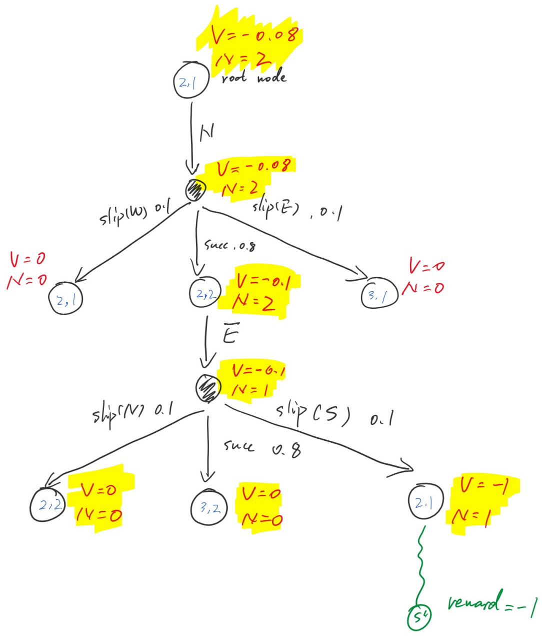

Figure 2: MCT after iteration 2

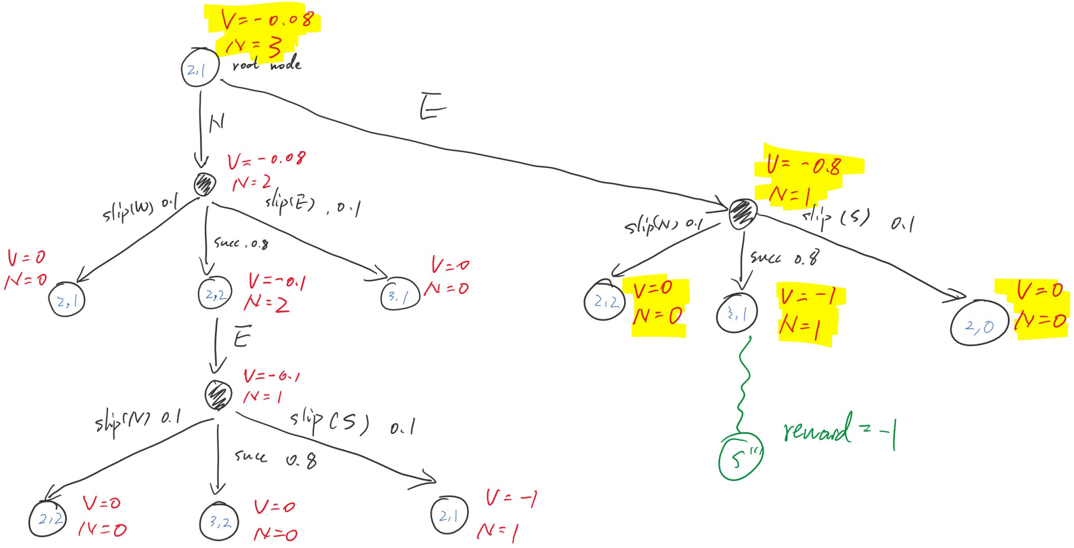

Figure 3: MCT after iteration 3

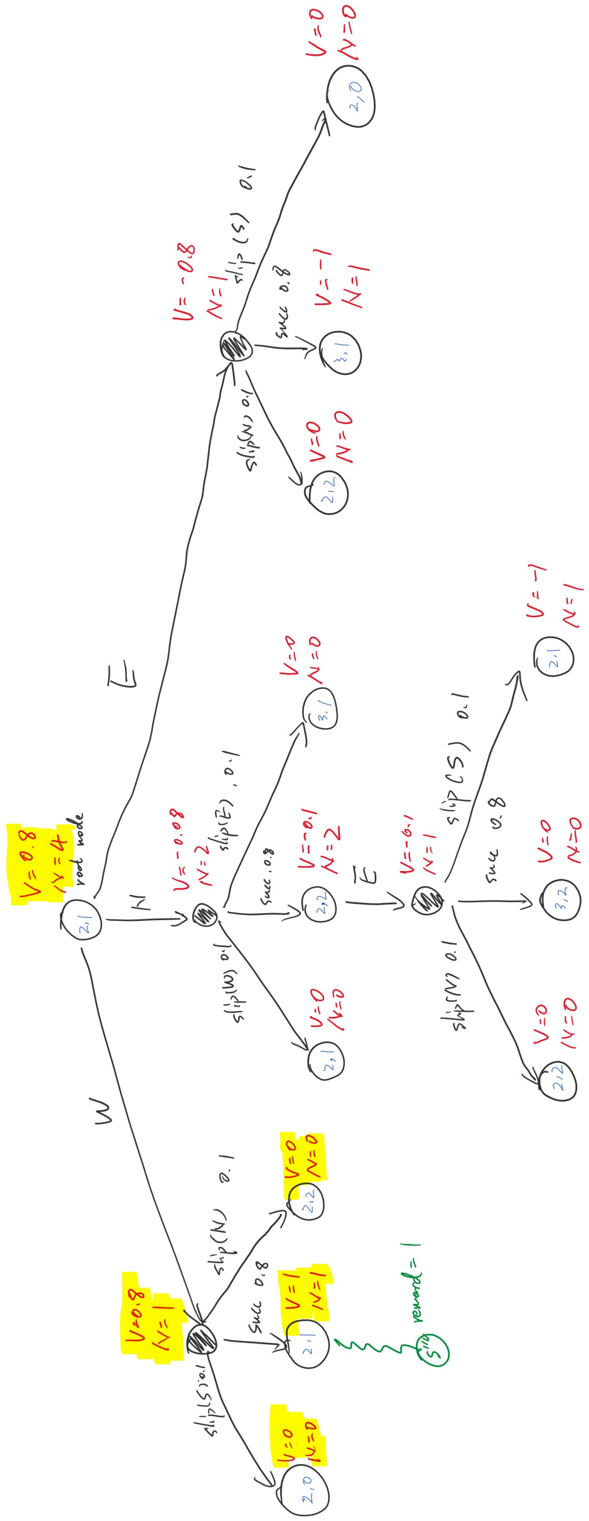

Figure 4: MCT after iteration 4

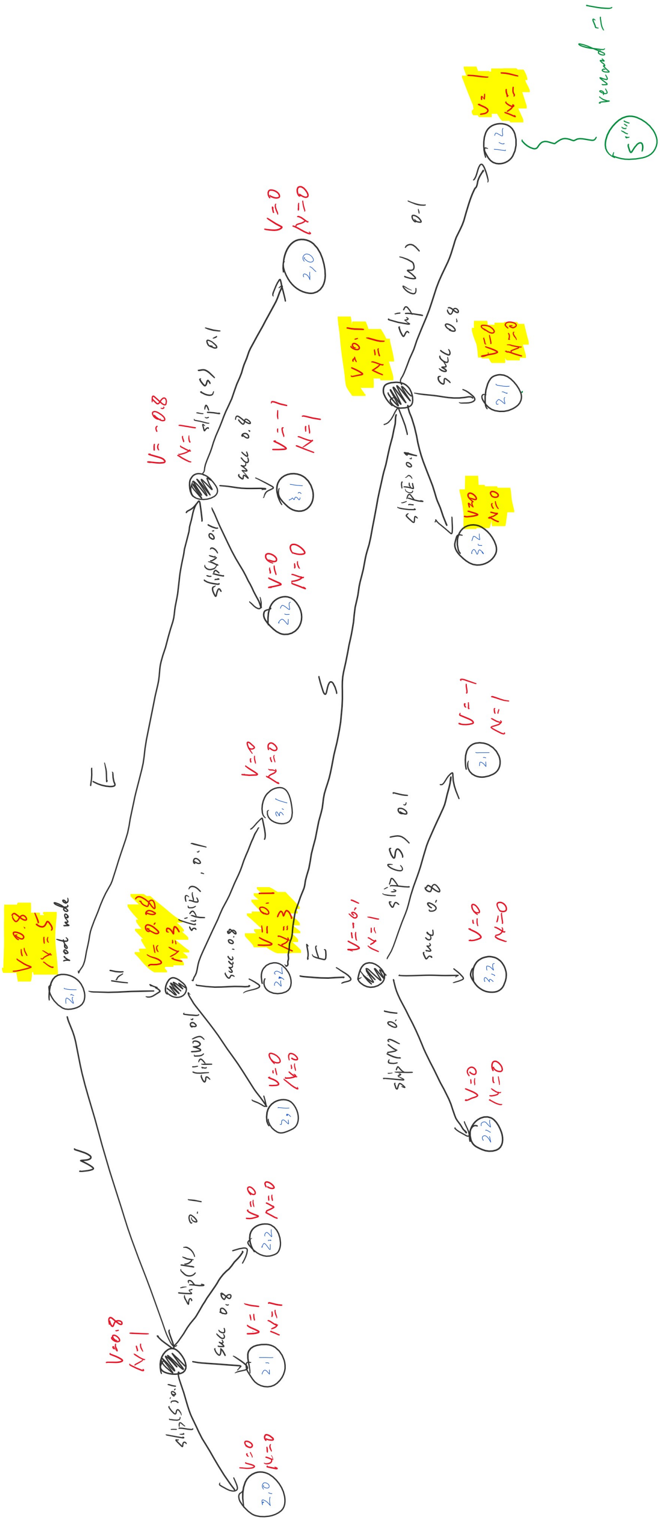

Figure 5: MCT after iteration 5