MDPs

Exercise 1

(a) **Environment:** Opponent's chess positions (Chess position: a 'instant picture' of the own pieces at particular turn)
States: Player's own pieces' chess position (dim: $num\_of\_diff\_pieces$ x $num\_of\_legal\_piece\_pos$, discrete)
Actions: legal moves for all the pieces $\approx$ 80, discrete
Reward: +1 for winning, -1 for losing, 0 for for draw. No intermediate rewards

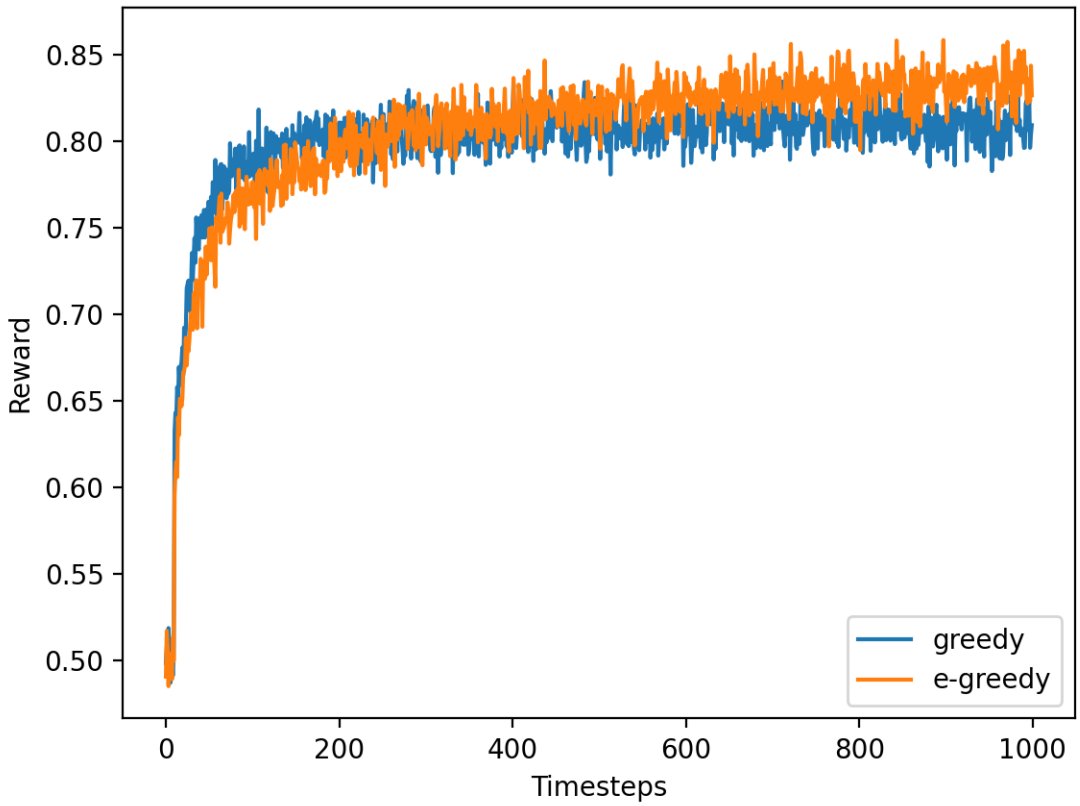
(b) **Environment:** 3D continuous space with coordinates of the object to pick up
States: Arm position in 8D space (continuous)
Actions: move the arm in one of the 8 dimensions (continuous) - back and forth, up and down, right and left, rotation
Reward: big positive reward for picking the object, e.g. 100, small negative rewards for each move, e.g. -1 to enforce efficiency.
The same for the placing procedure...

(c) **Environment:** 3D continuous space with possibly changing dynamics
States: position of each of the 4 rotors in 3D (continuous)
Actions: thrust power in each of the thrusters (4D, continuous)
Reward: Continuous reward for the time being on the same point on space

(d) Dialogue agent. Goal: accomplish a task the user asks for: book flight, provide information etc. in a dialogue setting. This is operating directly on utterance level, not on closed set of possible dialogue acts (templates) like most of the actual dialogue agents. The actual task can be accomplished after gathering enough knowledge from the user - turning unstructured text to structured information based on a knowledge graph, and reasoning om it.
Environment: User utterance
States: Discrete information - each user utterance produces new structured input - a edge or vertex on the knowledge graph. It's given from the environment and not agent's task to covert the utterance to structured input. Dimensions depend on the size of the knowledge graph
Actions: Pick a word from vocabulary - 40k (New states are encountered not after each action, but after building an utterance, when the agent hits the EOS token.)
Reward: Small negative for each dialogue turn, big positive reward if user marks task accomplished, big negative reward if human agent has to step in.

Exercise 2

(c) The ϵ-greedy method improves slower, but to a better average reward:



(d) We can use initialization with optimistic values to force exploration on the initial stages. Also Upper-Confidence-Bound action selection is shown to perform better than ϵ-greedy search, because it takes into account the uncertainty of the value of the chosen action - more frequently chosen actions are have lower uncertainty.