Dish Stacking with Reinforcement Learning

A domestic assistance robot needs to stack dishes from a dishwasher. It can grab and hold a dish from the washer or it can store a held dish in the cupboard. It can complete both perfectly with no chance of error. Unfortunately, the dishwasher is broken and hasn't dried any of the plates and the reward for storing dry plates in the cupboard (10) is larger than storing wet plates (5). The robot can attempt to dry any held plate but has a small chance of breaking the plate (pf =0.1) which carries a heavy cost (-20). The robot will also drop and break a held plate (-20) if it decides to grab a new one from the dishwasher. The robot now needs to try and determine which policy to choose for this task.

1. Model this dish stacking problem as a finite MDP. Write down the transition graph for this problem or write the reward and transition functions in matrix form.
2. Use the following algorithms to start trying to find an optimal policy for this MDP. You do not need to find the optimal policy, just perform two or three steps of each algorithm from scratch.
   A) Policy Iteration
   B) Value Iteration
   C) First-Visit Monte Carlo (for suggested samples see end of sheet)
   D) Every-Visit Monte Carlo
   E) SARSA Temporal Difference Learning
   F) Q-Learning Temporal Difference Learning
3. If I didn't know the probability of breaking plate during drying which of the 4 algorithms above should I use to find an optimal policy?
4. When using an ε-greedy algorithm, what is the effect of increasing ε?

For Monte-Carlo and TDL some starting samples could be:

s_0	a_1
s_1	a_2
s_2	a_1

s_0	a_0
s_1	a_2
s_2	a_0

s_0	a_0
s_1	a_2
s_2	a_1