POLICY ITERATION FOR MDPs

REPORT

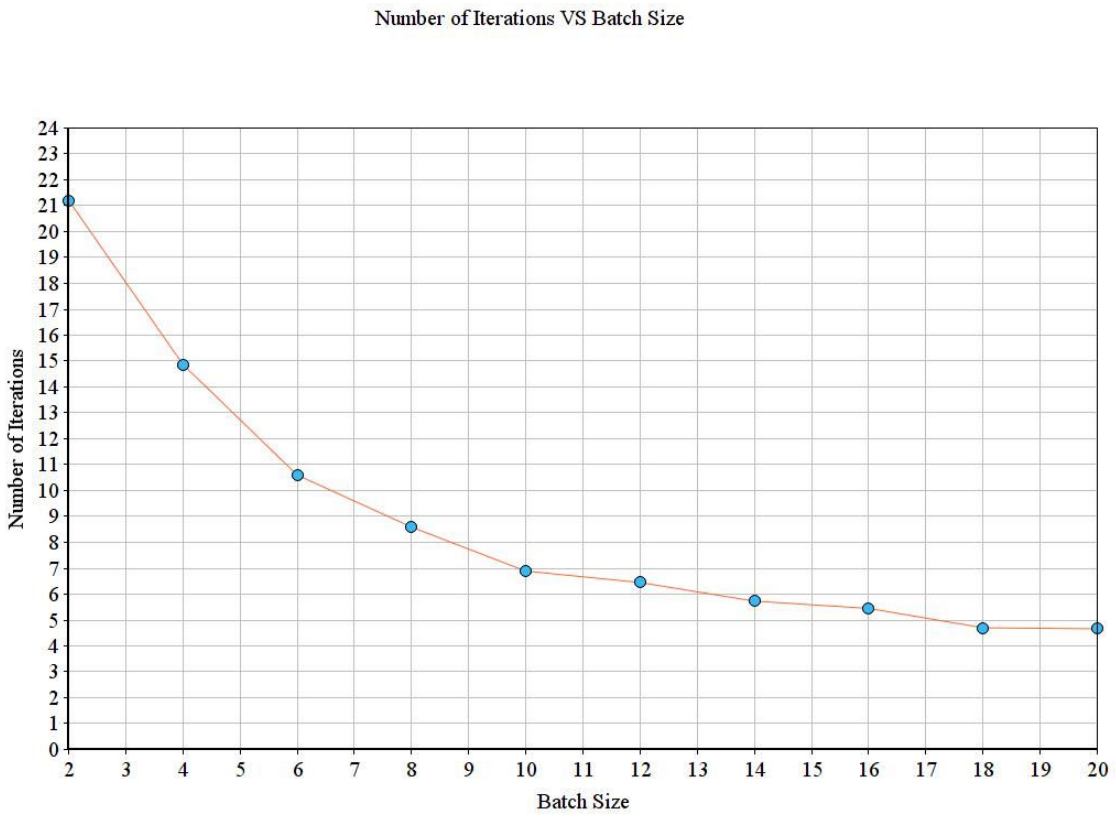
SUBMITTED BY:
130050010
SUYASH A BHATKAR

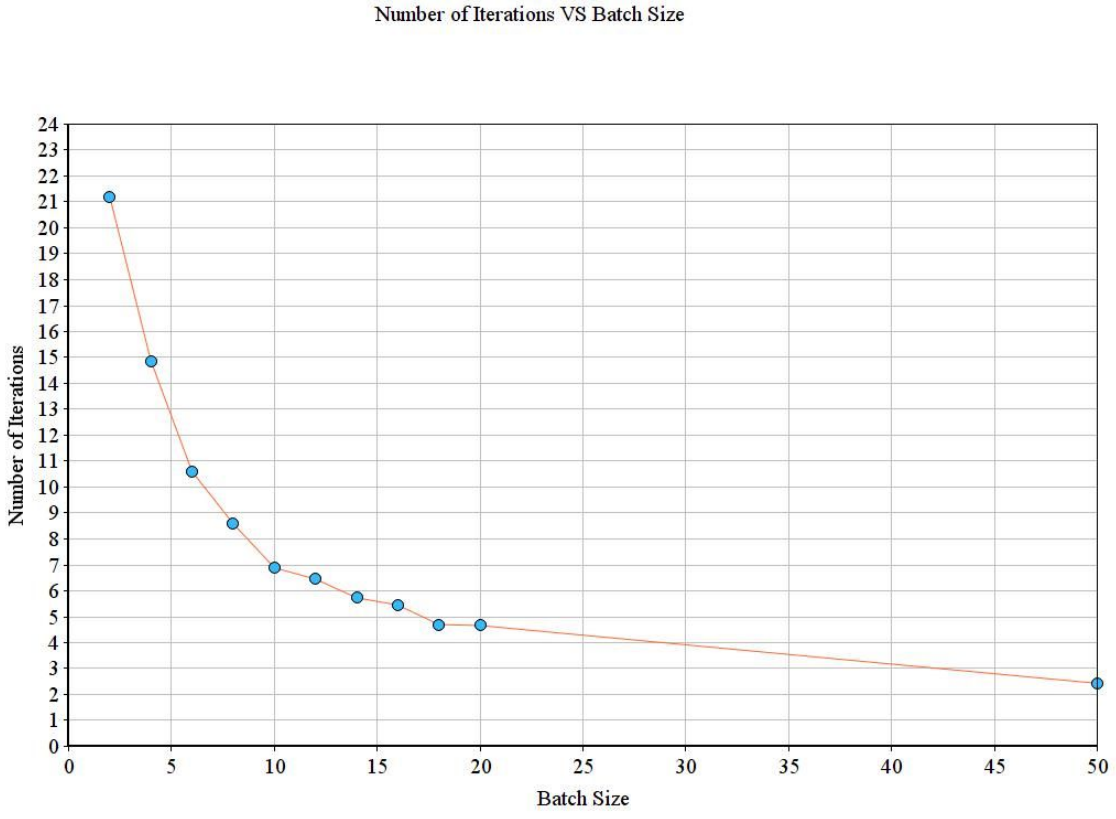
Average number of Iterations taken by the respective algorithms is given below:

POLICY ALGORITHM	AVERAGE NUMBER OF ITERATIONS
Howards Policy Iteration	2.43
Randomised Policy Iteration	7.32
Batch Switching with Size = 2	21.21
Batch Switching with Size = 4	14.87
Batch Switching with Size = 6	10.6
Batch Switching with Size = 8	8.59
Batch Switching with Size = 10	6.89
Batch Switching with Size = 12	6.45
Batch Switching with Size = 14	5.73
Batch Switching with Size = 16	5.45
Batch Switching with Size = 18	4.7
Batch Switching with Size = 20	4.66
Batch Switching with Size >= 50	2.43

*These are obtained for the same set of MDPs. (generated as described below)

Graph for the number of Iterations in Batch Switching PI VS Batch Size is given below:





Process used to generate MDPs:

I generated the MDPs using the file “MDPGenerator.sh” in combination with “MDPGenerator.py” submitted with the report. Basically, MDPGenerator.py generates one MDP using random number generator in python. I have used the same convention as provided to generate them. The Transition Matrix values are normalised with respect to a state and action so, $\sum_{s' \varepsilon S}$ T[state][action][s’] = 1. Basically the agent has to go to some other state in the next step. So, the sum of the probabilities will be 1. The script “MDPGenerator.sh” loops around 100 times and creates the 100 MDPs by running “MDPGenerator.py” 100 times and stores the MDPs in the folder “$pwd/GeneratedMDPs/MDPk.txt” for all k between 1 and 100. In short, the MDPs can be generated by running “MDPGenerator.sh”.

Results obtained by PI Variants:

Howard’s Policy Iteration changes the improvable states right away, the Mansour and Singh’s Randomised Policy Iteration changes a subset of states whereas the Batch-Switching Policy Iteration changes the rightmost batch of the policy. The results could be predicted for a 2-action pair, but for more than 2 actions it is not as trivial.

The Howard’s PI is the simplest of them all and does not take more than 3 steps for any MDP of state size = 50. The average number of iterations is 2.43 which means that this easily is the best algorithm, compared to the others given the number of actions is 2.

The Randomised PI improves only a subset of the actions. This one is bound to be slower than the HPI because it improves less number of states per iteration than HPI.

The Batch Switching PI was an interesting one because a pattern is observed in its iterations. As you go on increasing the batchsize, the number of iterations taken by the program decreases because more states are given the chance to improve with each iteration. As we can see from the above graph that when the batchsize becomes greater than or equal to the number of states, no distinction remains between the Howard’s PI and Batch Switching PI because there is only one batch. So, all the states are improved in one iteration.

The clear winner here in a 2-action system is Howard’s PI but we don’t know how it might fare against the others in a multi-action(more than 2) MDP.