CS-747 Assignment 2 Report
Arun Verma, 154190002

Steps to run experiment:

Please find ***experiment*** directory in submitted code.
Note: Assuming your present directory is ../experiment/
To generate MDP instances
Run command: ***python mdpGenerator.py***
mdpGenerator.py file will generate 100 MDP instances with 50 states and 2 actions. The discount factor is chosen uniformly random between 0 and 1. The rewards are generated uniformly in between -1 and 1 by uniformly random. The transient probability are generated by following algorithm:
genertate_Transient_Probability:

1. Generate a random number (R_i) between 0 and 1 for first K-1 states for state S and action A
2. Transient probability of going state i form state any S = (R_i)/Σ(R_i) where i in (1, K-1)
3. The transient probability of going state K form state any S = 1 - [(R_i)/Σ(R_i)] where i in (1, K-1)

Comparison of different policy iteration algorithms:

Run command: ***python comparePI.py > output/compareResult.txt***
Observations: For solving **100** MDP instances given in ***../experiment/input*** directory with **50** states and **2** actions for **random seed 89** and **batch size 10**. As Howard's PI improves improvable states in one shot while Randomised PI chooses a subset of improvable states randomly and in Batch-Switching PI fixed batch size improvable states (from right to left). As number of improved states are less in each iteration, then it takes more iterations to converge the PI algorithm which is evident from below table.

Algorithm	Average number of Iterations
Howard's PI	**2.44**
Mansour and Singh's Randomised PI	**5.8**
Mansour and Singh's Batch-switching PI	**6.83**

Batch-switching PI with different batch size:

Run command: ***python batchPI.py > output/batchResult.txt***
Observation: As batch size is increased, number of iterations decreases because in each iteration, we are improving more states than in smaller batch size. For batch size = Number of states, Batch Switching PI algorithm is equivalent to Howard's PI algorithm.