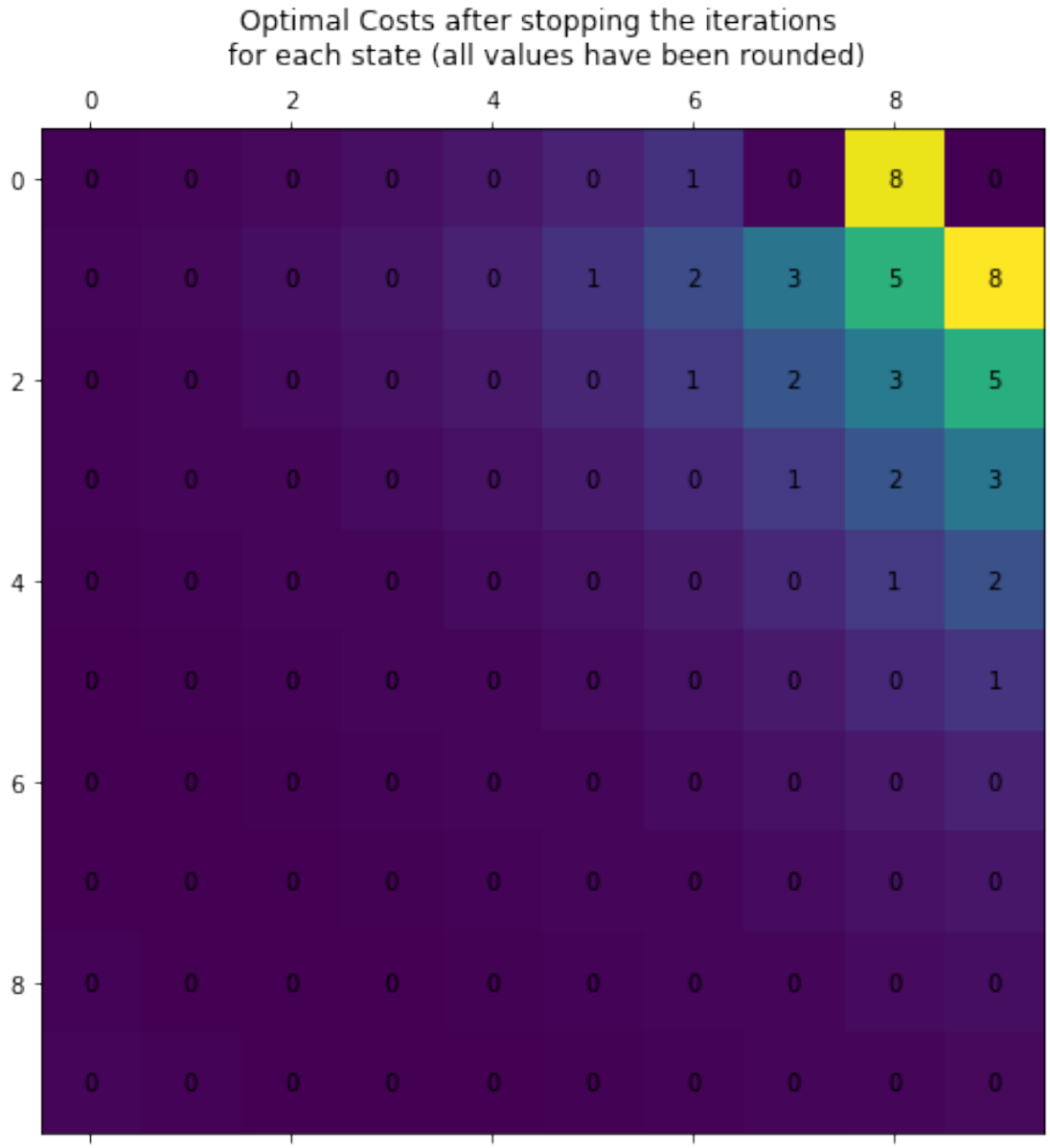
Assignment 4

Siddharth Nayak EE16B073

18th Oct 2018

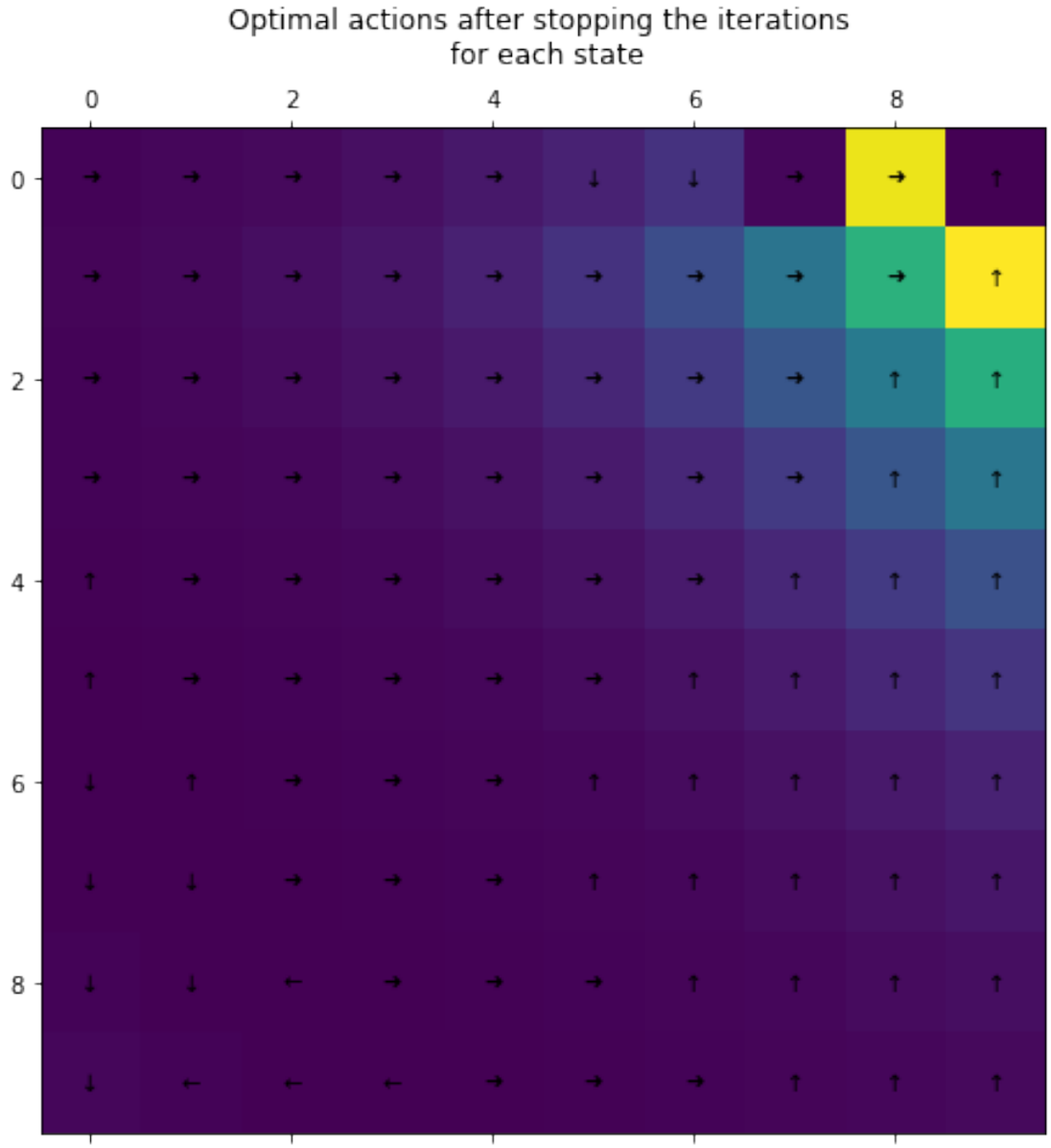
1 Grid World

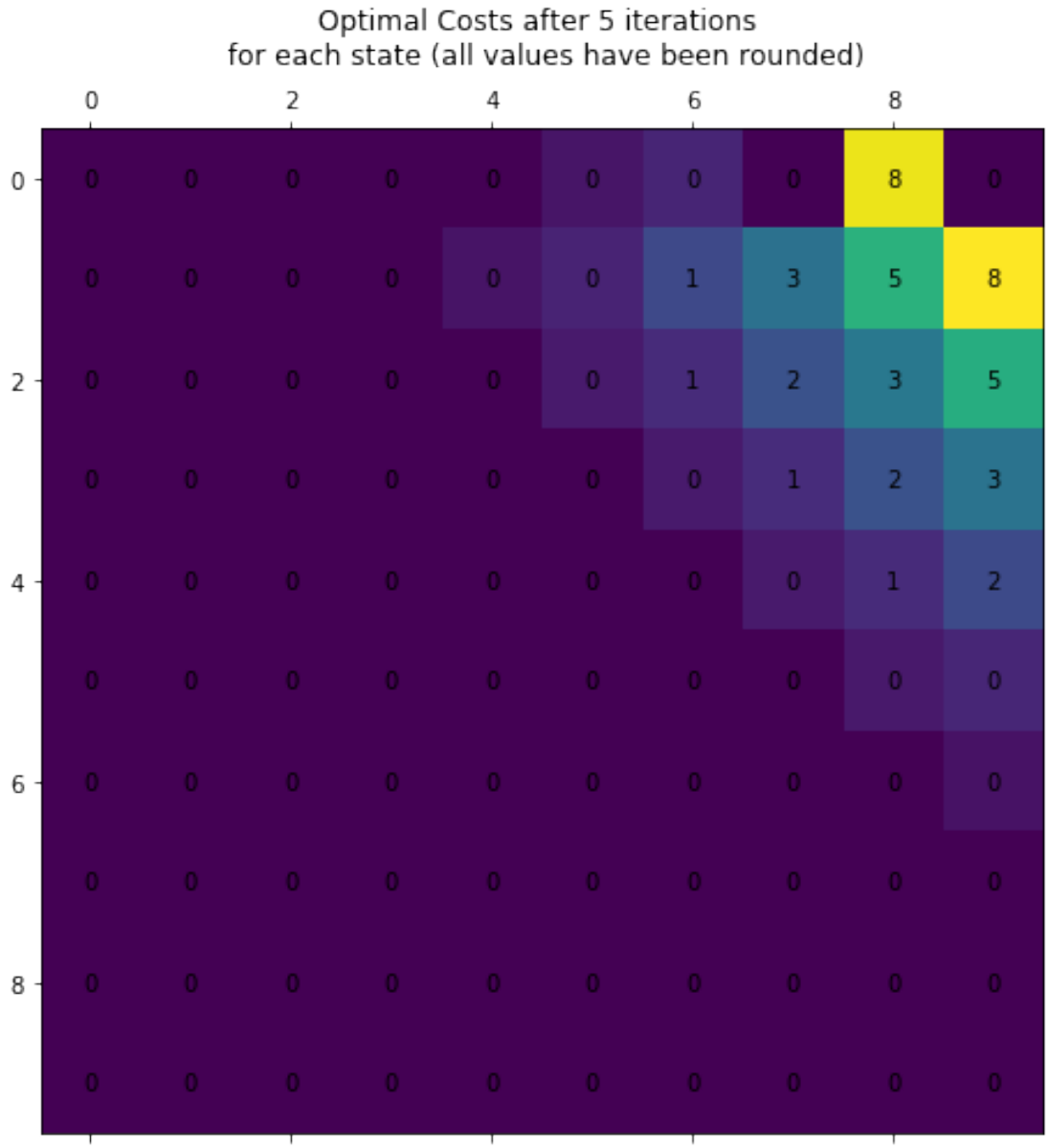
1.1 Value Iteration

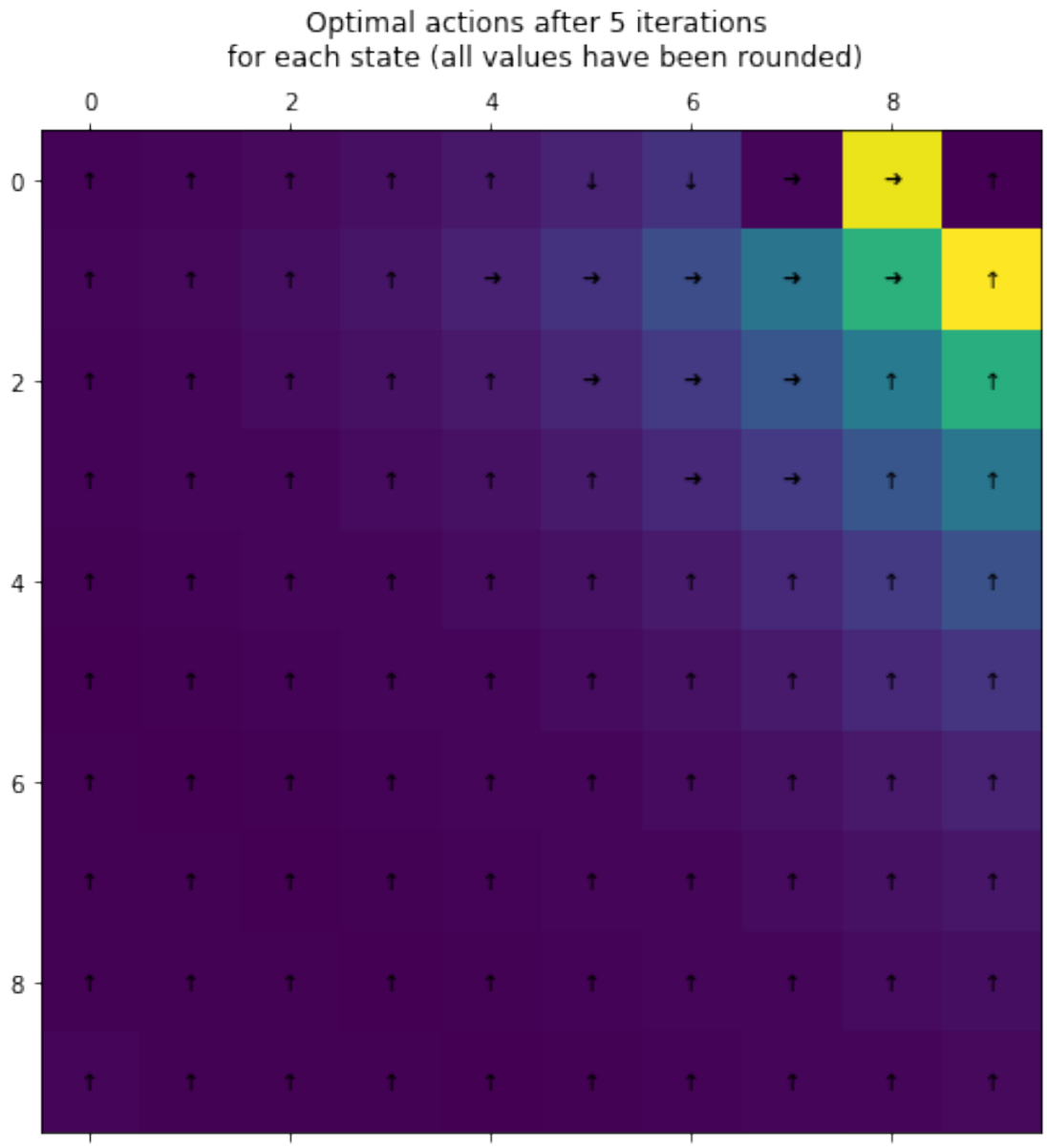


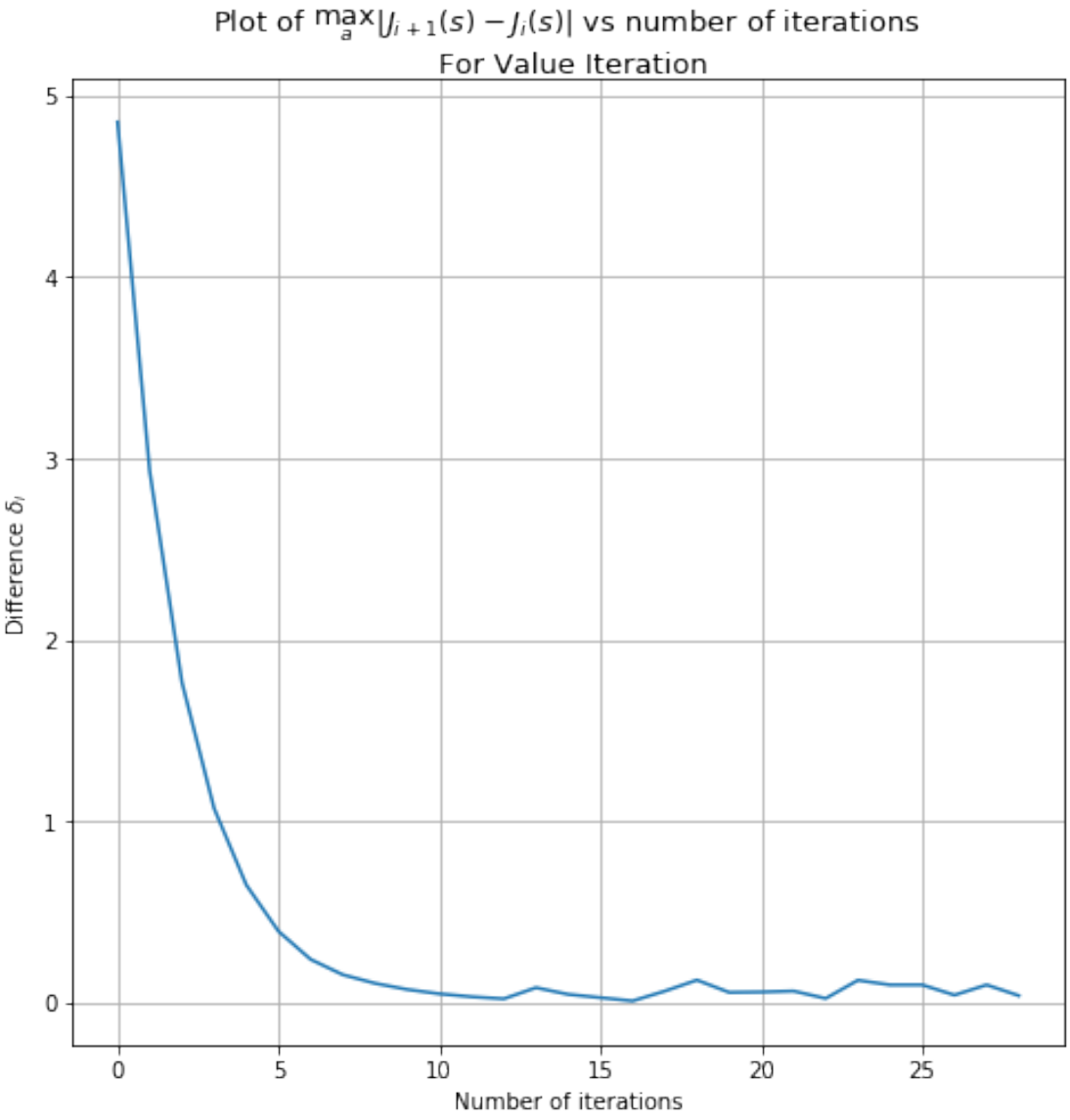
1.1.1 Explanation of policy obtained

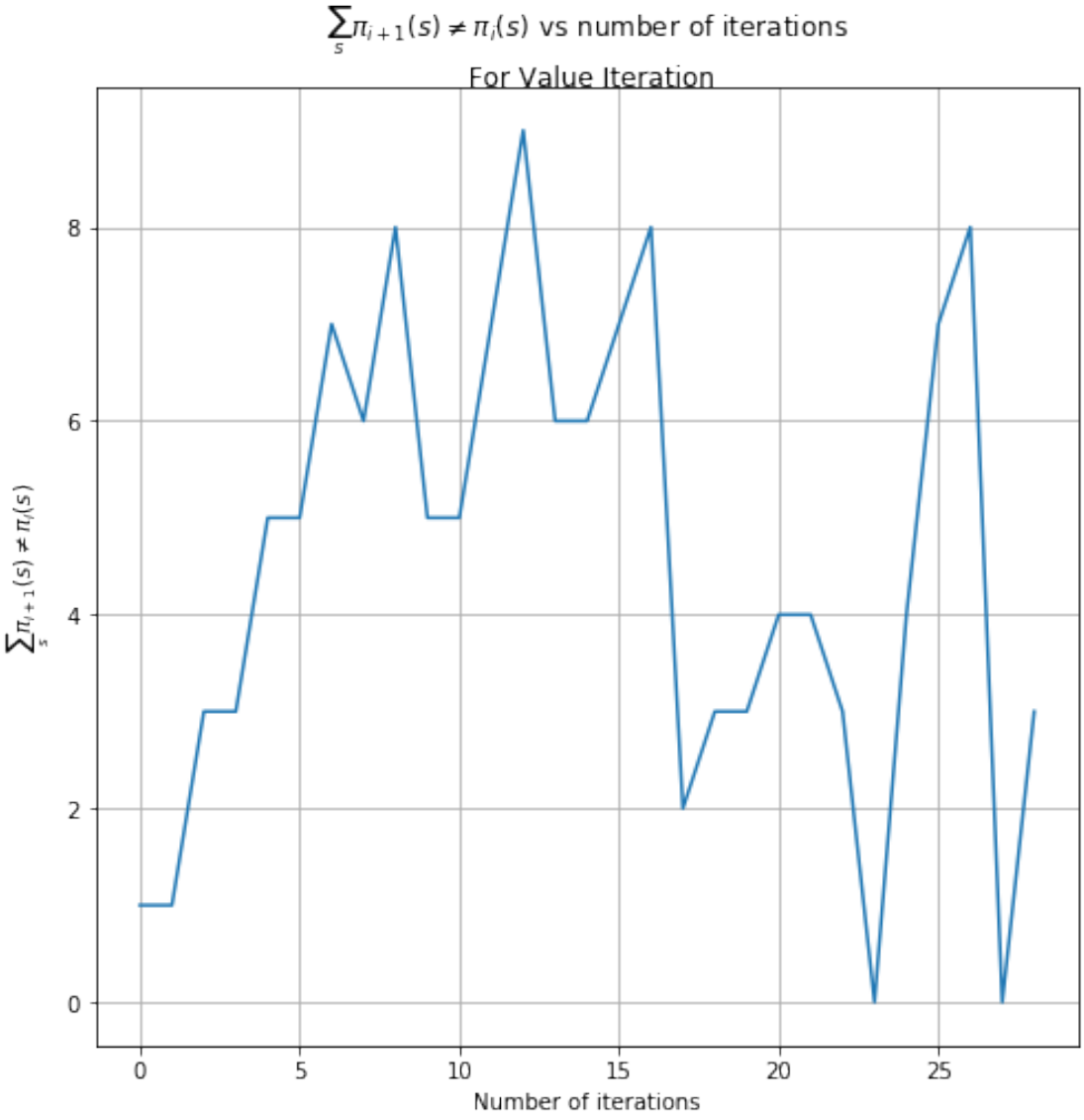
Value Iteration: All the actions around the Goal1 point into the goal.
None of the arrows point into Orange IN as it leads us far away from the goal.
All the actions point into Gray IN as it takes us closer towards Goal1



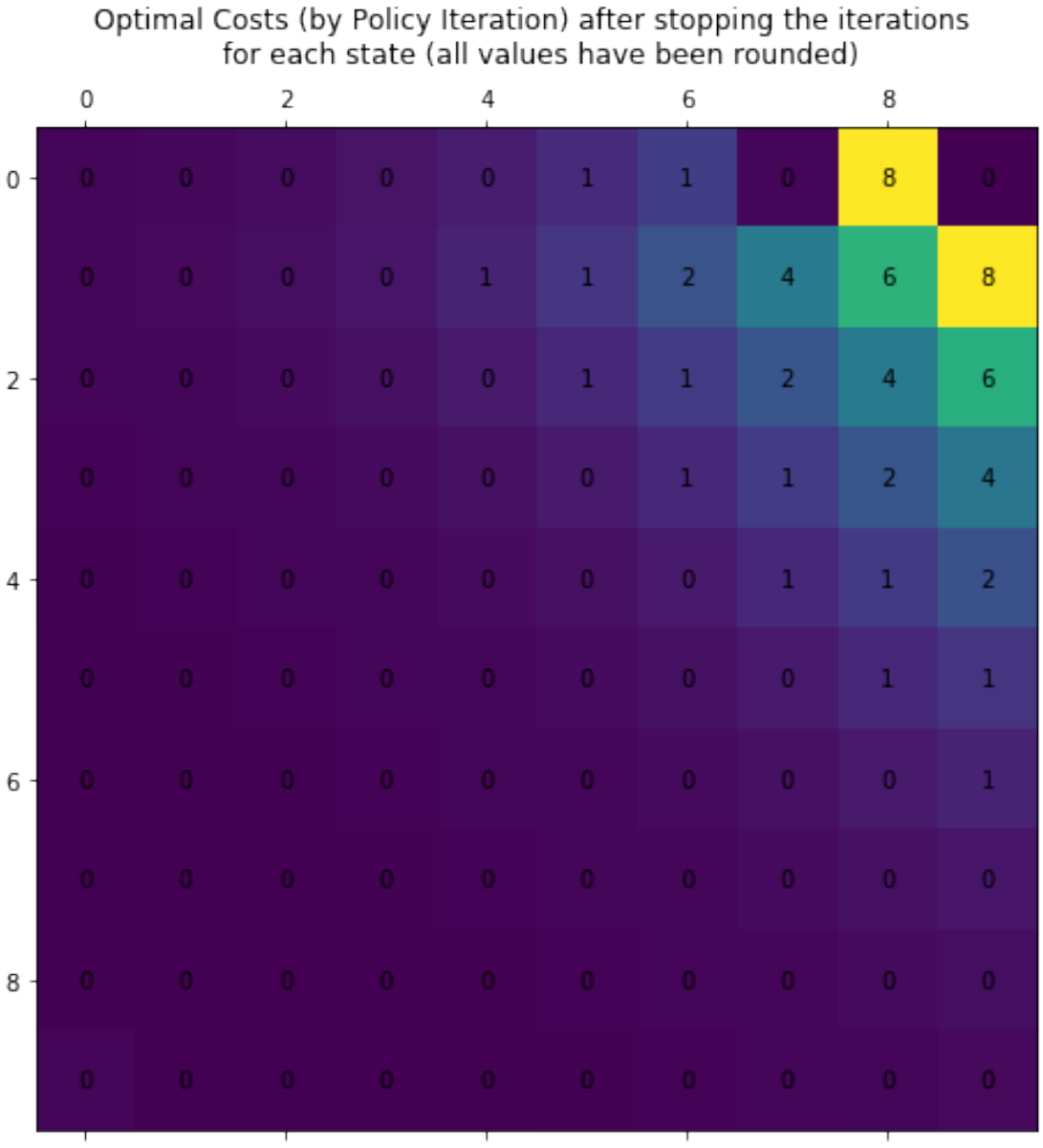

Optimal Costs after 5 iterations
for each state (all values have been rounded)
0
2
4
6
8
0 0 0 0 0 0 0 0 8 0
0 0 0 0 0 0 1 3 5 8
0 0 0 0 0 0 1 2 3 5
0 0 0 0 0 0 0 1 2 3
0 0 0 0 0 0 0 0 1 2
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0


Optimal actions after 5 iterations
for each state (all values have been rounded)
0
2
4
6
8


Plot of $\max_a |J_{i+1}(s) - J_i(s)|$ vs number of iterations
For Value Iteration
Difference δ_i
Number of iterations
0
1
2
3
4
5
0
5
10
15
20
25

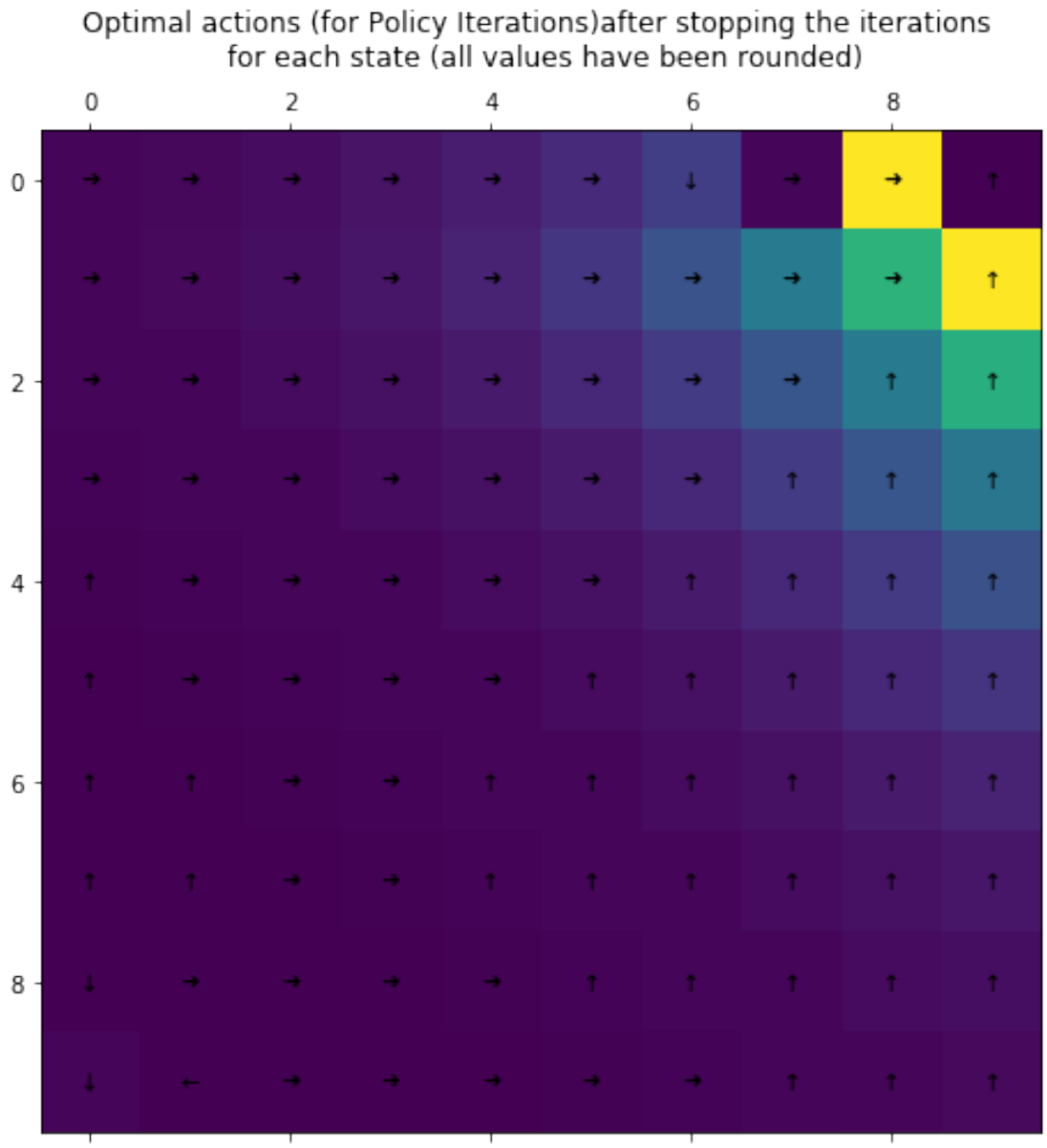

$\sum_s \pi_{i+1}(s) \neq \pi_i(s)$ vs number of iterations
For Value Iteration
$\sum_s \pi_{i+1}(s) \neq \pi_i(s)$
0
2
4
6
8
0
5
10
15
20
25
Number of iterations

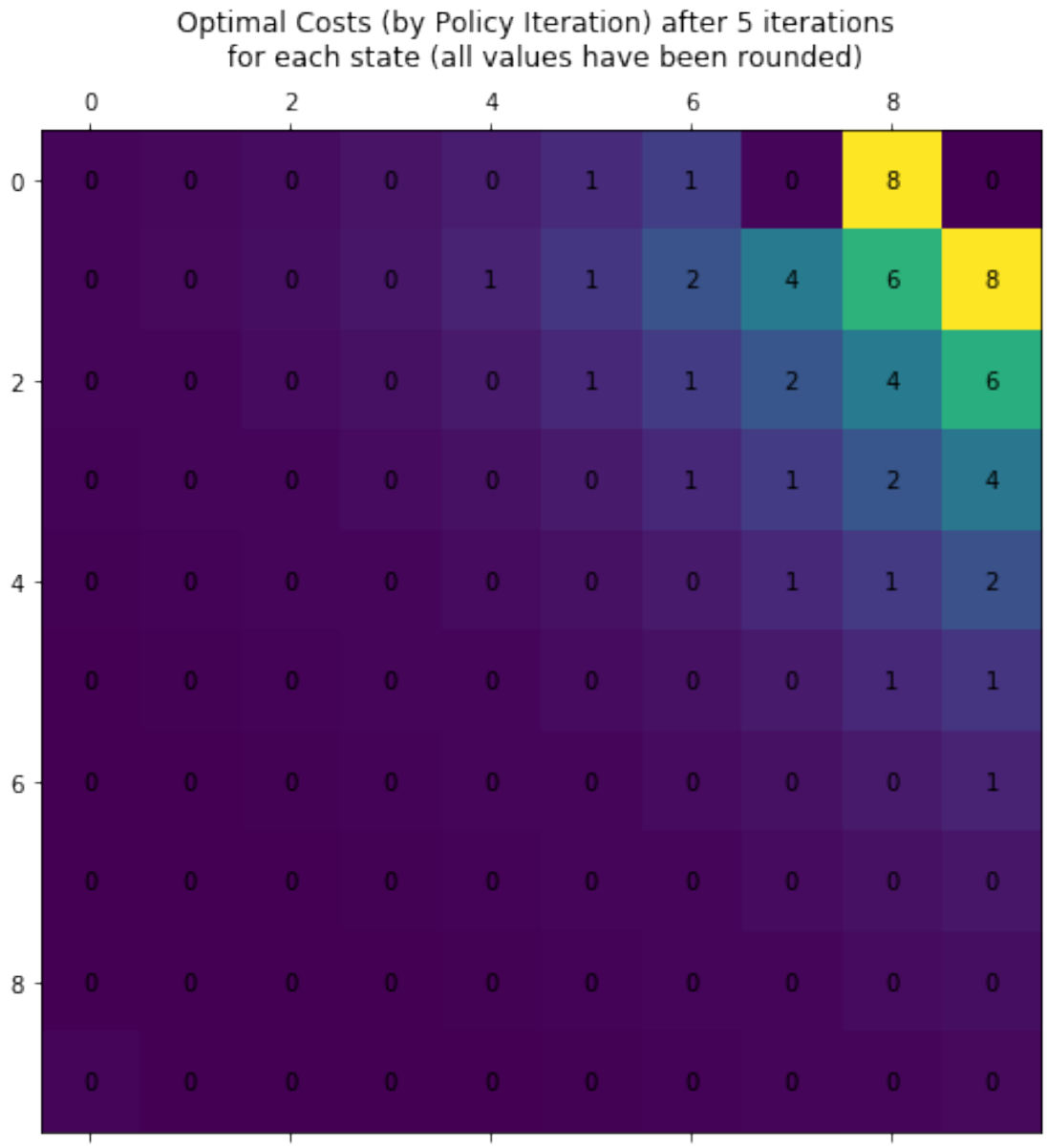
1.1.2 Policy Iteration

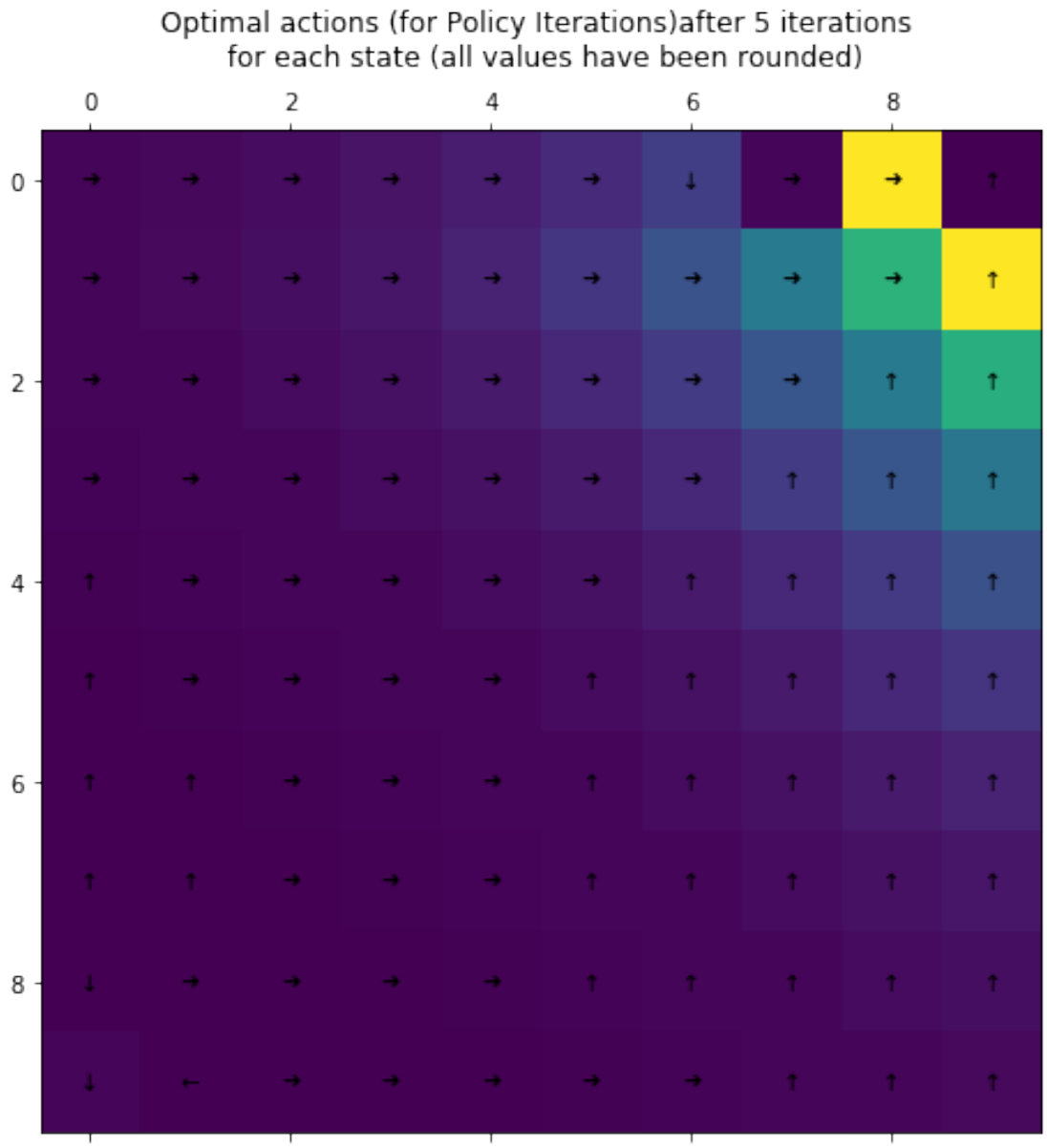


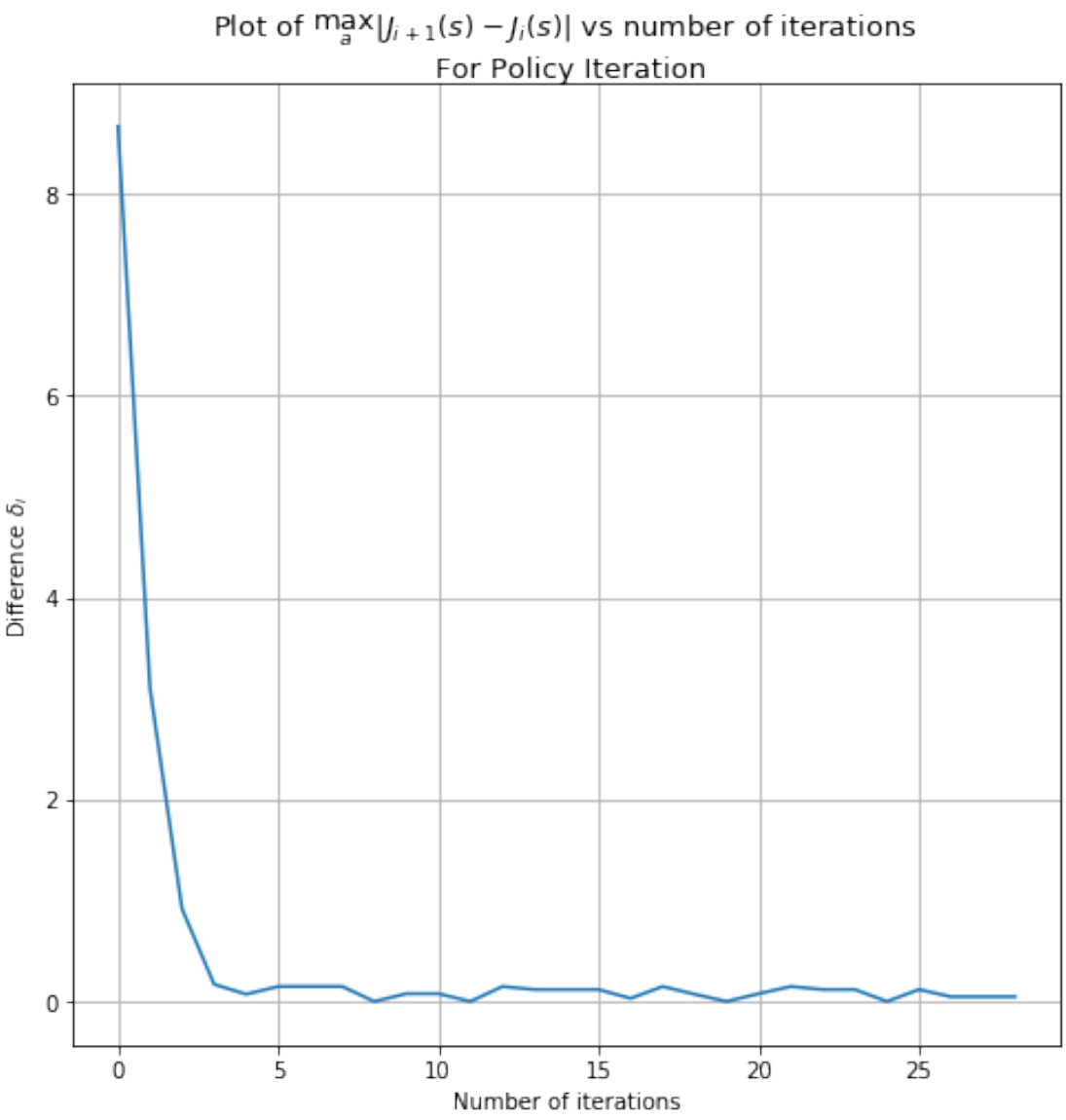
1.1.3 Explanation of policy obtained

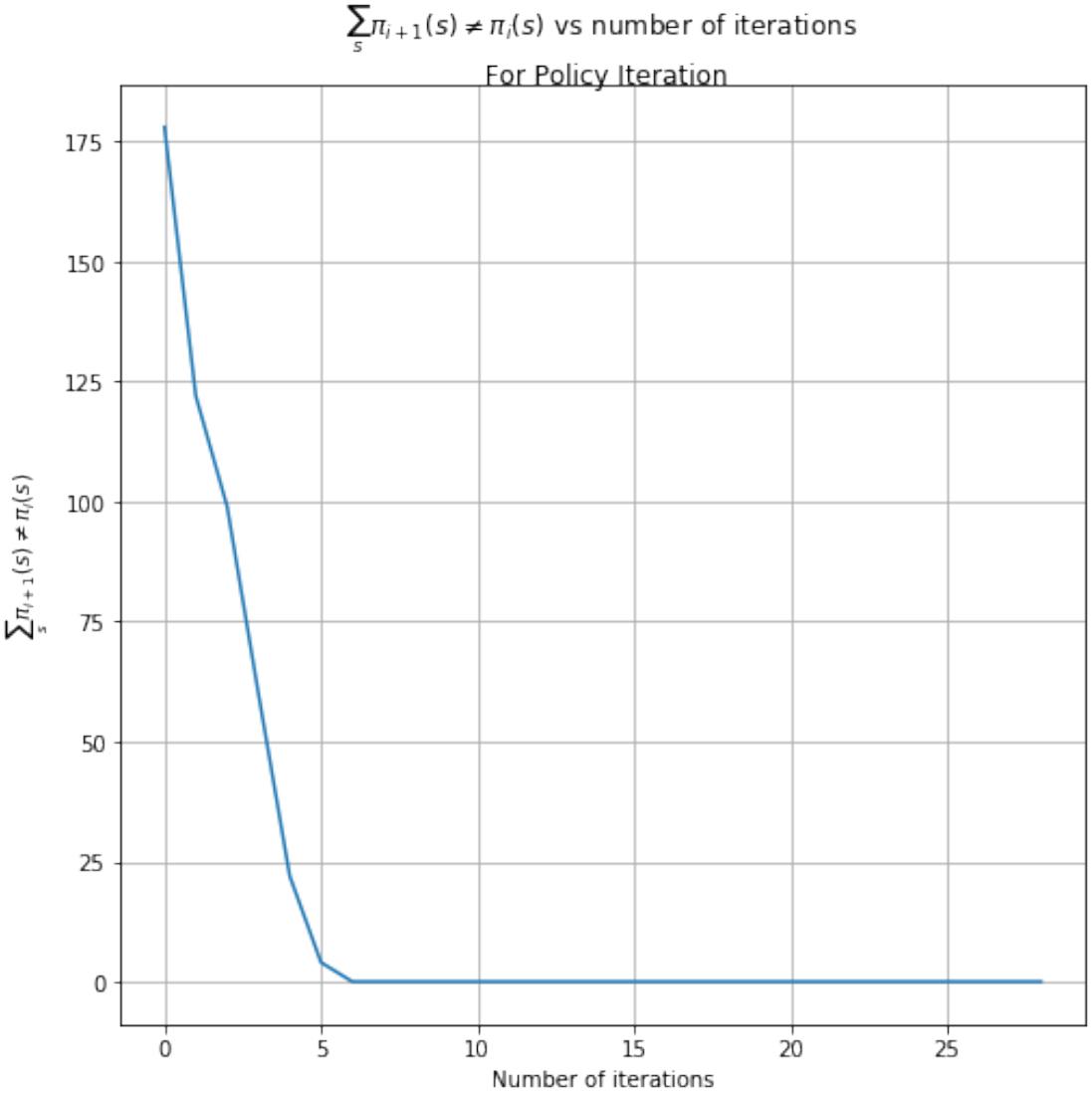
Policy Iteration: All the actions around the Goal1 point into the goal.
None of the arrows point into Orange IN as it leads us far away from the goal.
All the actions point into Gray IN as it takes us closer towards Goal1


Optimal actions (for Policy Iterations)after stopping the iterations
for each state (all values have been rounded)
0
2
4
6
8
0
2
4
6
8


Optimal Costs (by Policy Iteration) after 5 iterations
for each state (all values have been rounded)
0 2 4 6 8
0 0 0 0 0 0 1 1 0 8 0
0 0 0 0 0 1 1 2 4 6 8
2 0 0 0 0 0 1 1 2 4 6
0 0 0 0 0 0 1 1 2 4
4 0 0 0 0 0 0 0 1 1 2
0 0 0 0 0 0 0 0 1 1
6 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0
8 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0

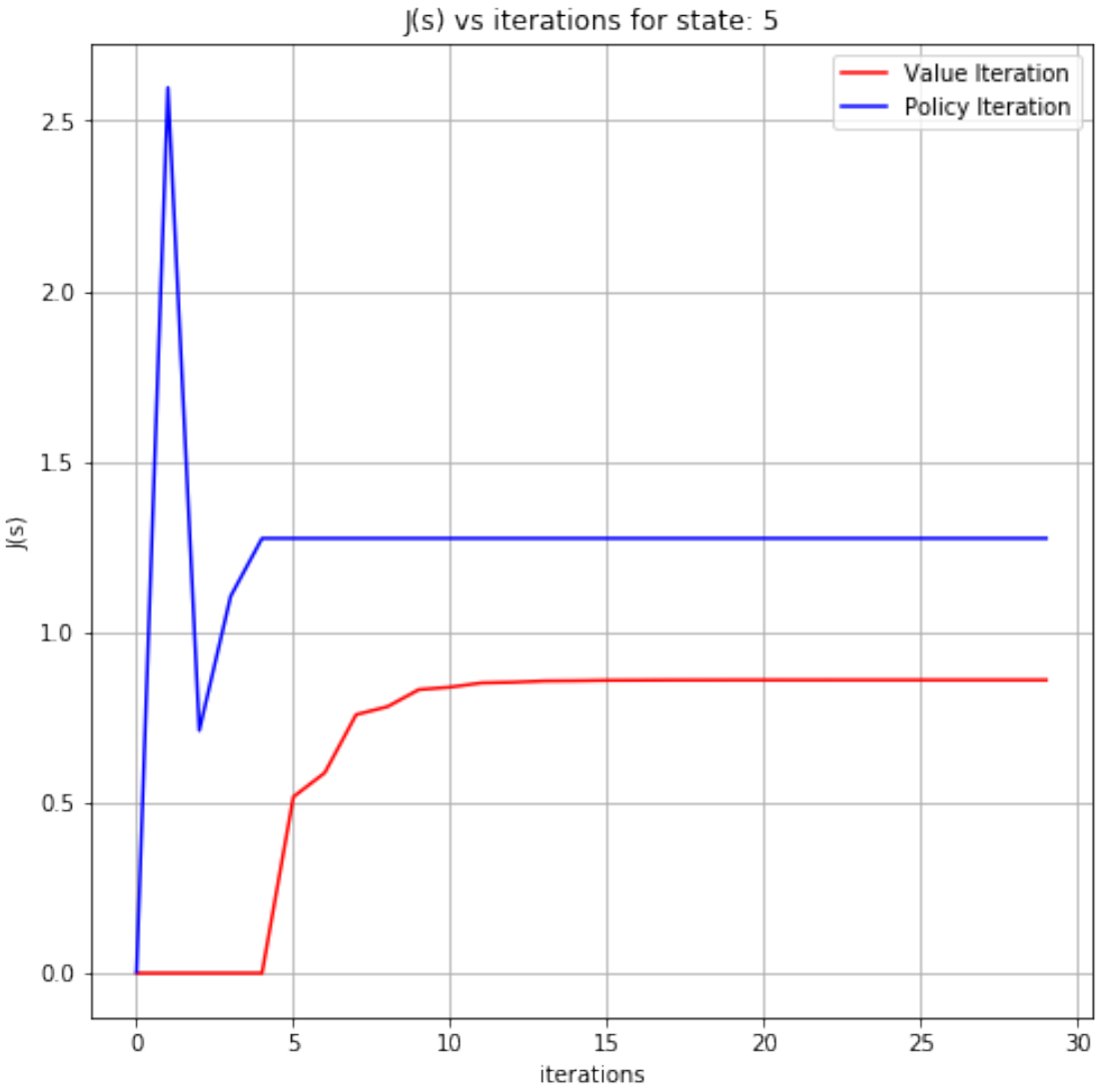

Optimal actions (for Policy Iterations)after 5 iterations
for each state (all values have been rounded)
0
2
4
6
8
0
2
4
6
8

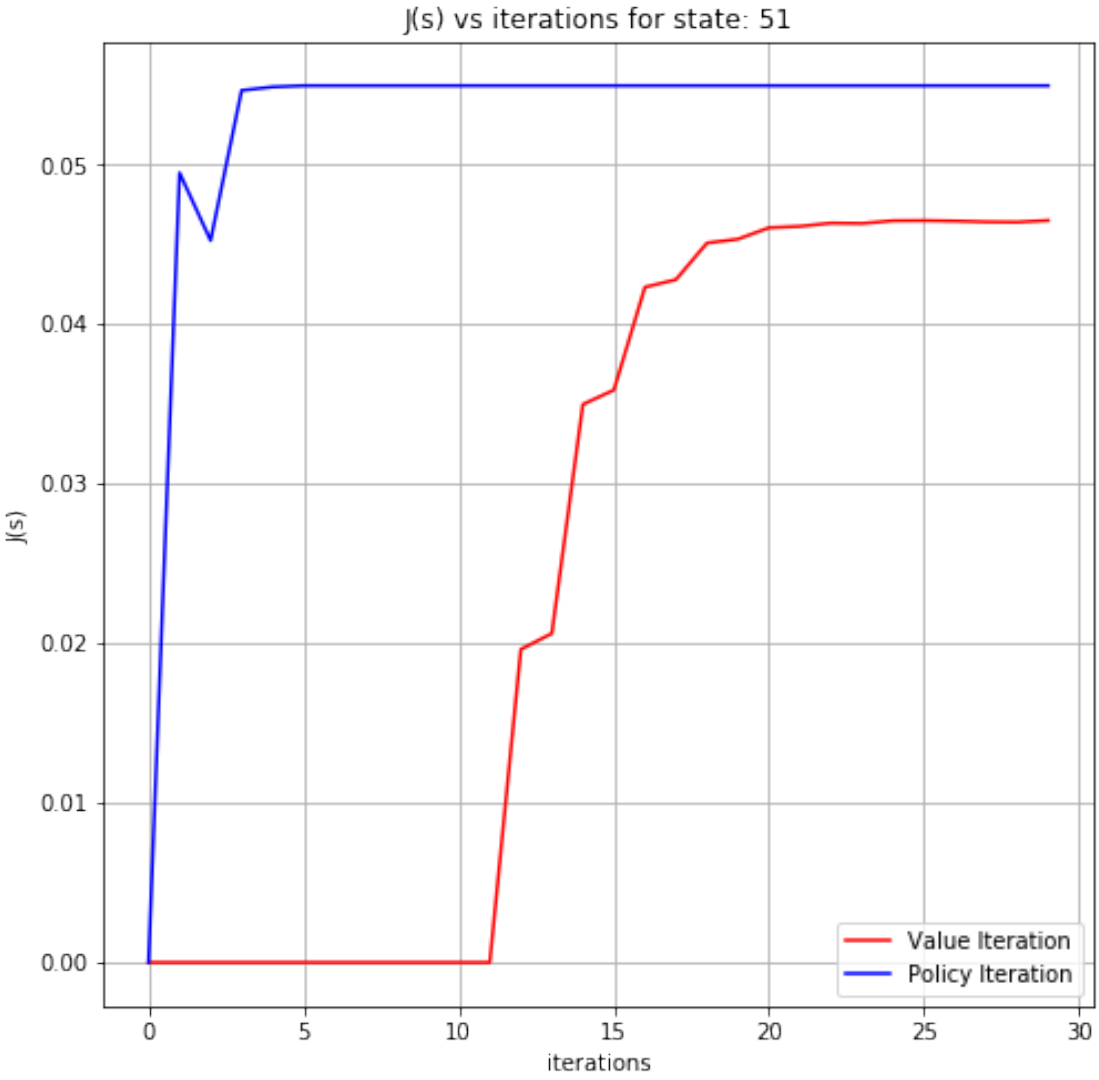

Plot of $\max_a |J_{i+1}(s) - J_i(s)|$ vs number of iterations
For Policy Iteration
Difference δ_i
Number of iterations
0
2
4
6
8
0
5
10
15
20
25

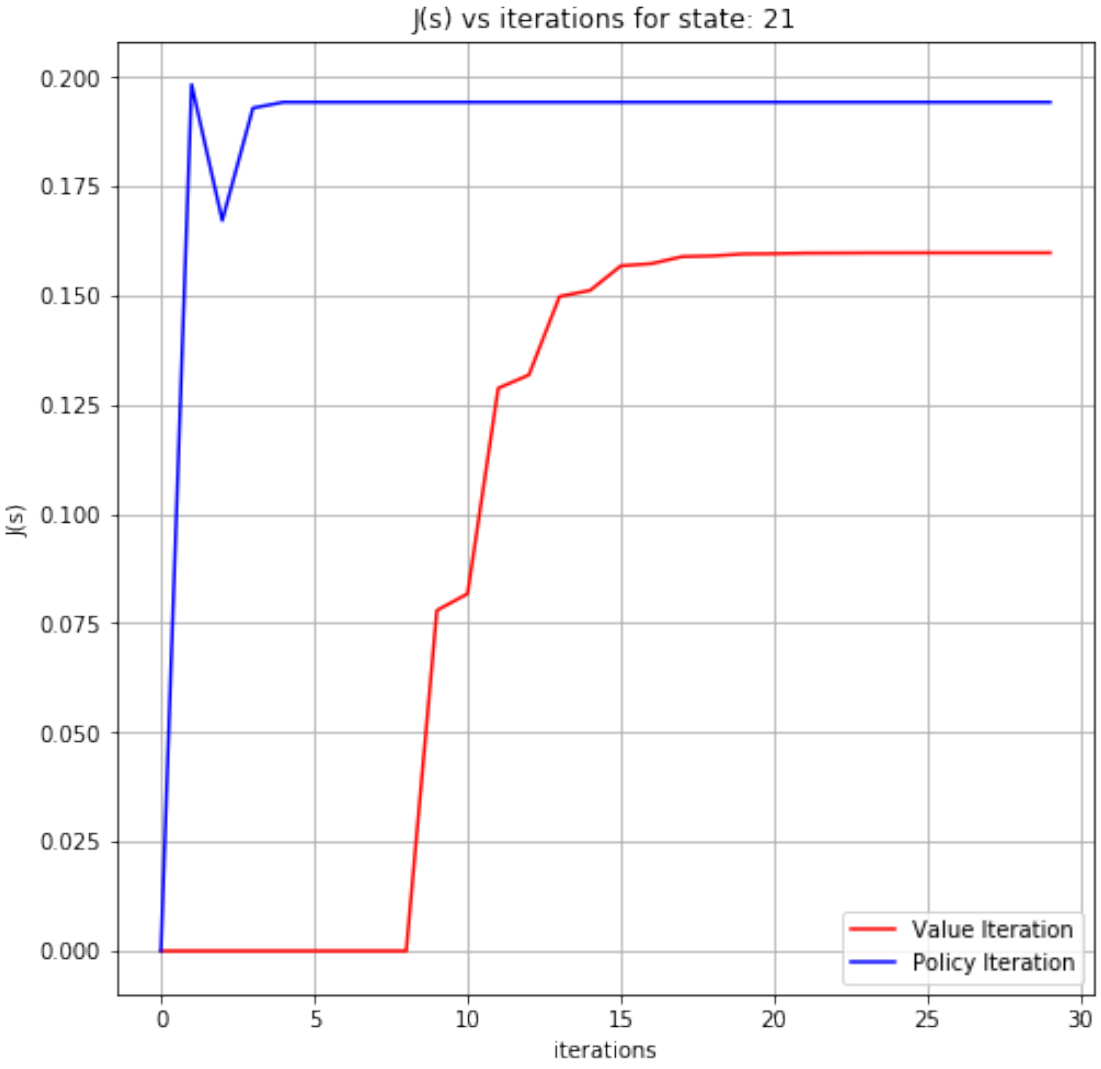


Comparing Policy iteration and value iteration:

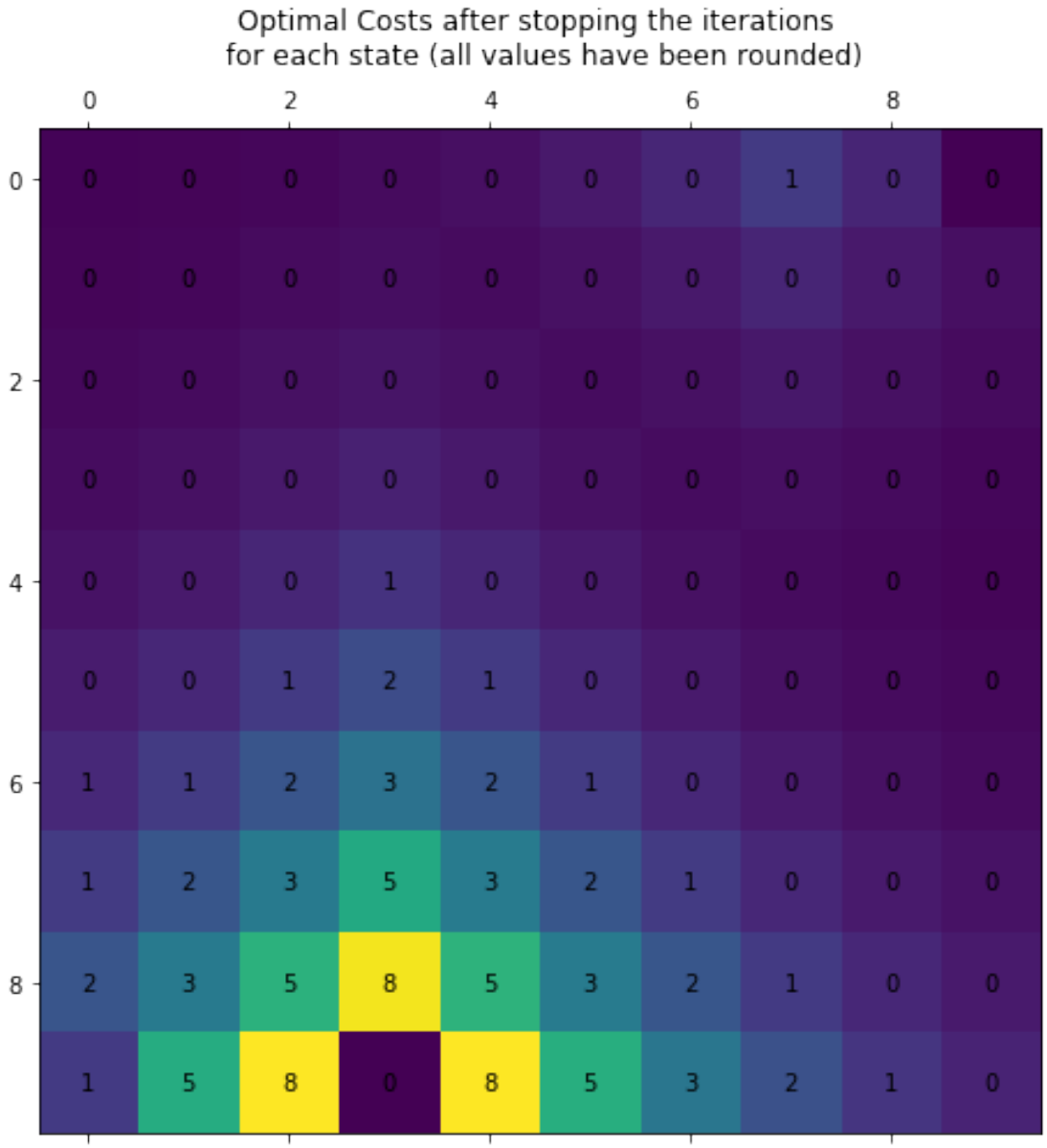
With the three plots given below it is evident that policy iteration converges faster. This is because we are updating the policies after each iteration giving us a different J(s) whereas in value iteration the convergence is theoretically after infinite steps.


J(s) vs iterations for state: 5
Value Iteration
Policy Iteration
J(s)
iterations
0.0
0.5
1.0
1.5
2.0
2.5
0
5
10
15
20
25
30


J(s) vs iterations for state: 51
J(s)
iterations
0.00
0.01
0.02
0.03
0.04
0.05
0
5
10
15
20
25
30
Value Iteration
Policy Iteration

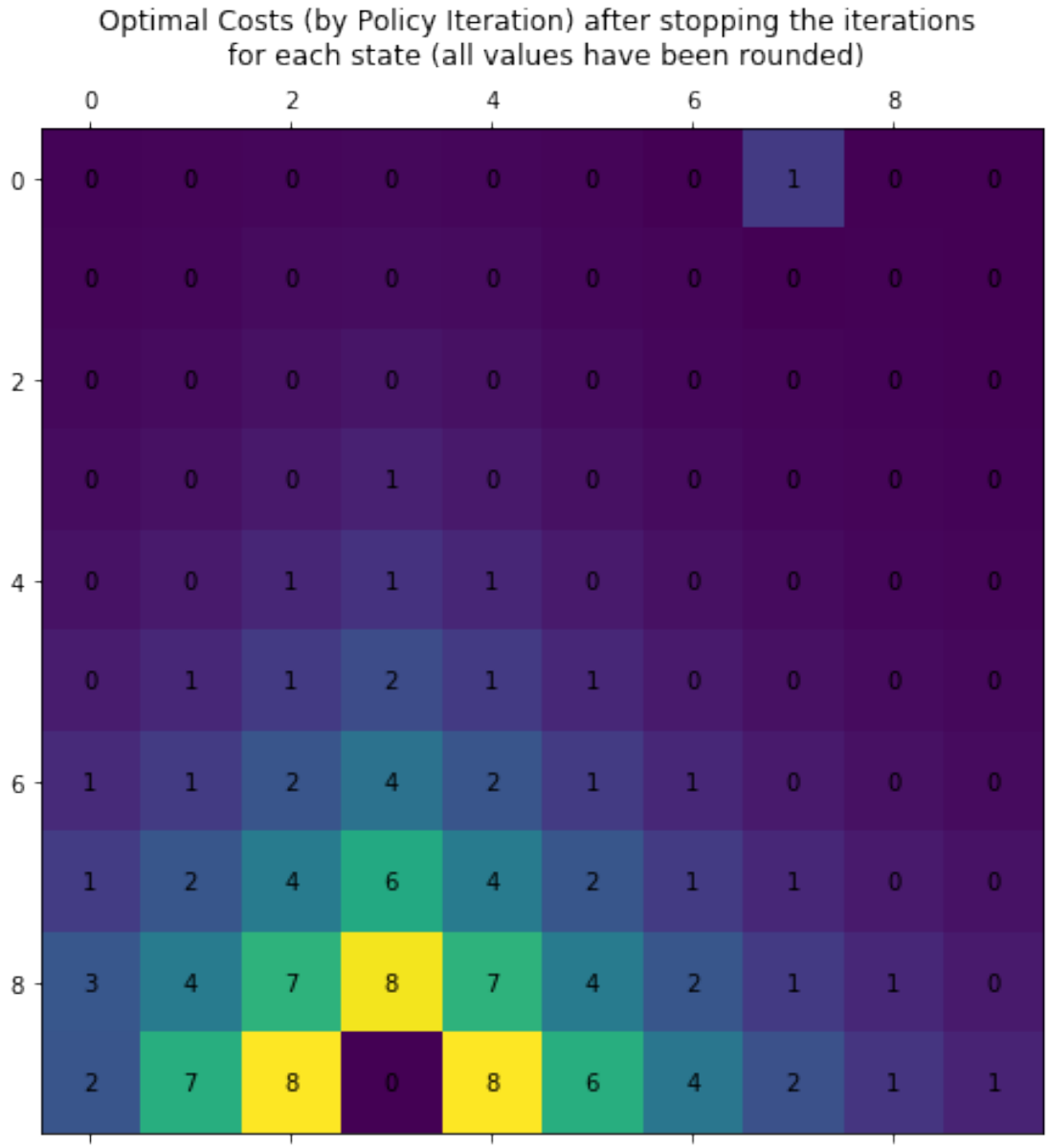

J(s) vs iterations for state: 21
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
J(s)
0
5
10
15
20
25
30
iterations
Value Iteration
Policy Iteration

Goal 2



1.1.4 Explanation of policy obtained

Value Iteration: All the actions around the Goal2 point into the goal.
None of the arrows point into Gray IN as it leads us far away from the goal.
All the actions point into Orange IN as it takes us closer towards Goal1



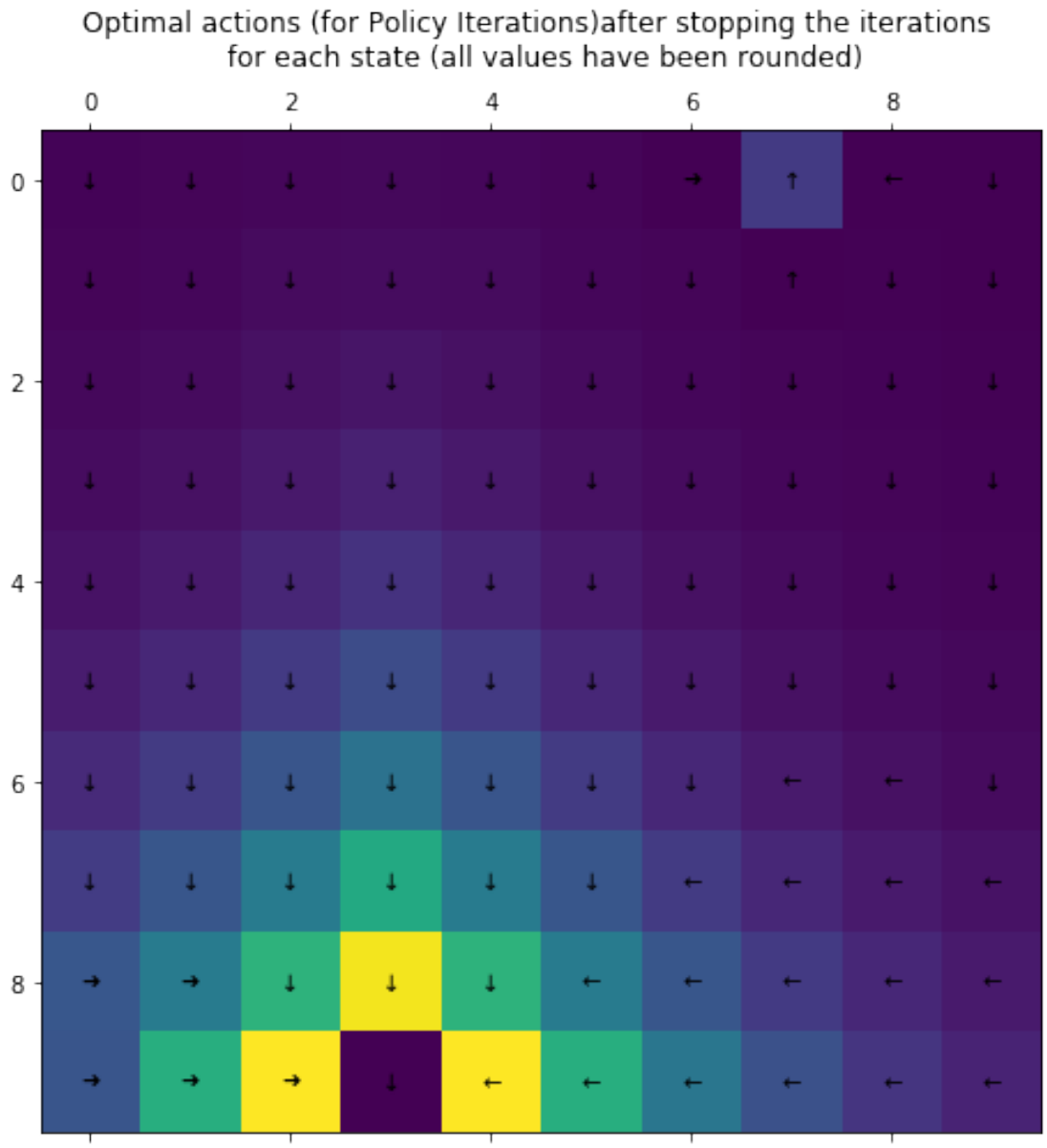
1.1.5 Explanation of policy obtained

Policy Iteration: All the actions around the Goal2 point into the goal.

None of the arrows point into Gray IN as it leads us far away from the goal.

All the actions point into Orange IN as it takes us closer towards Goal1

Note: In all the grids showing the actions and costs the values/actions plotted in Goals and IN boxes are not valid as we cannot take action in that state.



2 Taxi

2.1 Policy Iteration

In the given problem we have the following actions:

- Cruise the streets looking for a passenger.
- Go to the nearest taxi stand and wait in line.
- Wait for a call from the dispatcher (this is not possible in town B because of poor reception).

2.1.1 Policy Iteration (Part 1):

Here the values of β are varied from 0 to 0.95 with step size 0.05 and the optimal rewards and Optimal Actions are shown below:

Evident from the table, we can see that the optimal rewards increase as we increase the value of β

Also the optimal actions initially (for small β) is to take action:1, i.e.Cruise the streets looking for a passenger. And for large values of β it is optimal to take action:2, i.e. Go to the nearest taxi stand and wait in line.

Optimal Costs obtained by Policy Iteration for each of the states for different values of alpha

Out[51]:

	alpha	A	B	C
0	0.00	8.000000	16.000000	7.000000
1	0.05	8.511527	16.400260	7.498869
2	0.10	9.076506	16.856369	8.050865
3	0.15	9.704456	17.385855	8.665495
4	0.20	10.407268	18.377651	9.354637
5	0.25	11.200000	19.500414	10.133333
6	0.30	12.102002	20.782180	11.020921
7	0.35	13.138573	22.259614	12.042683
8	0.40	14.343434	23.981600	13.232323
9	0.45	15.762563	26.014797	14.635803
10	0.50	17.460317	28.452525	16.678788
11	0.55	19.529649	31.429582	19.572894
12	0.60	22.109974	35.148160	23.207424
13	0.65	26.698885	39.925990	27.900087
14	0.70	32.982802	46.292620	34.180406
15	0.75	41.807514	55.201235	43.001544
16	0.80	55.079365	68.558201	56.269841
17	0.85	77.246512	90.811701	78.433456
18	0.90	121.653471	135.306276	122.836903
19	0.95	255.022908	268.764619	256.202849

Optimal actions obtained by Policy Iteration for each of the states for different values of alpha

Out[52]:

	alpha	A	B	C
0	0.00	1	1	1
1	0.05	1	1	1
2	0.10	1	1	1
3	0.15	1	2	1
4	0.20	1	2	1
5	0.25	1	2	1
6	0.30	1	2	1
7	0.35	1	2	1
8	0.40	1	2	1
9	0.45	1	2	1
10	0.50	1	2	1
11	0.55	1	2	2
12	0.60	1	2	2

13	0.65	1	2	2
14	0.70	1	2	2
15	0.75	1	2	2
16	0.80	2	2	2
17	0.85	2	2	2
18	0.90	2	2	2
19	0.95	2	2	2

2.2 Modified Policy Iteration

Choosing $m_k = 5$ we get the optimal cost of
A : 121.6497741
B : 135.3025785
C : 122.83320606
and optimal actions
A:2
B:2
C:2

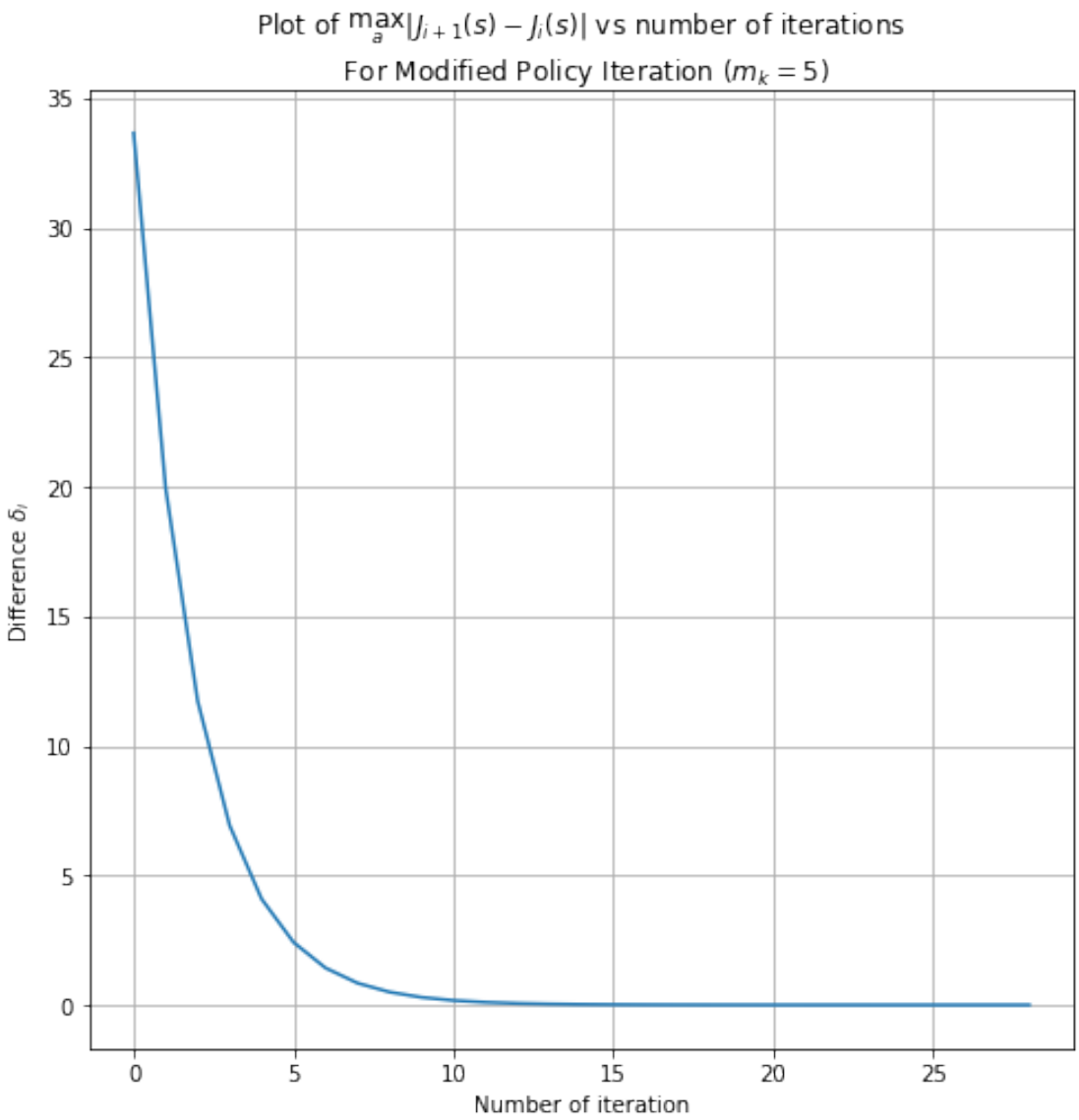
Also the plot of $\delta_i = \max_a |J_{i+1}(s) - J_i(s)|$ vs number of iterations For Modified Policy Iteration $(m_k = 5)$ shows that the algorithm converges ($\delta_i < 0.001$) after 20 iterations.

Choosing $m_k = 5$ we get the optimal cost of
A : 121.65347112
B : 135.30627552
C : 122.83690308
and optimal actions
A:2
B:2
C:2

Also the plot of $\delta_i = \max_a |J_{i+1}(s) - J_i(s)|$ vs number of iterations For Modified Policy Iteration $(m_k = 10)$ shows that the algorithm converges ($\delta_i < 0.001$) after 11 iterations.

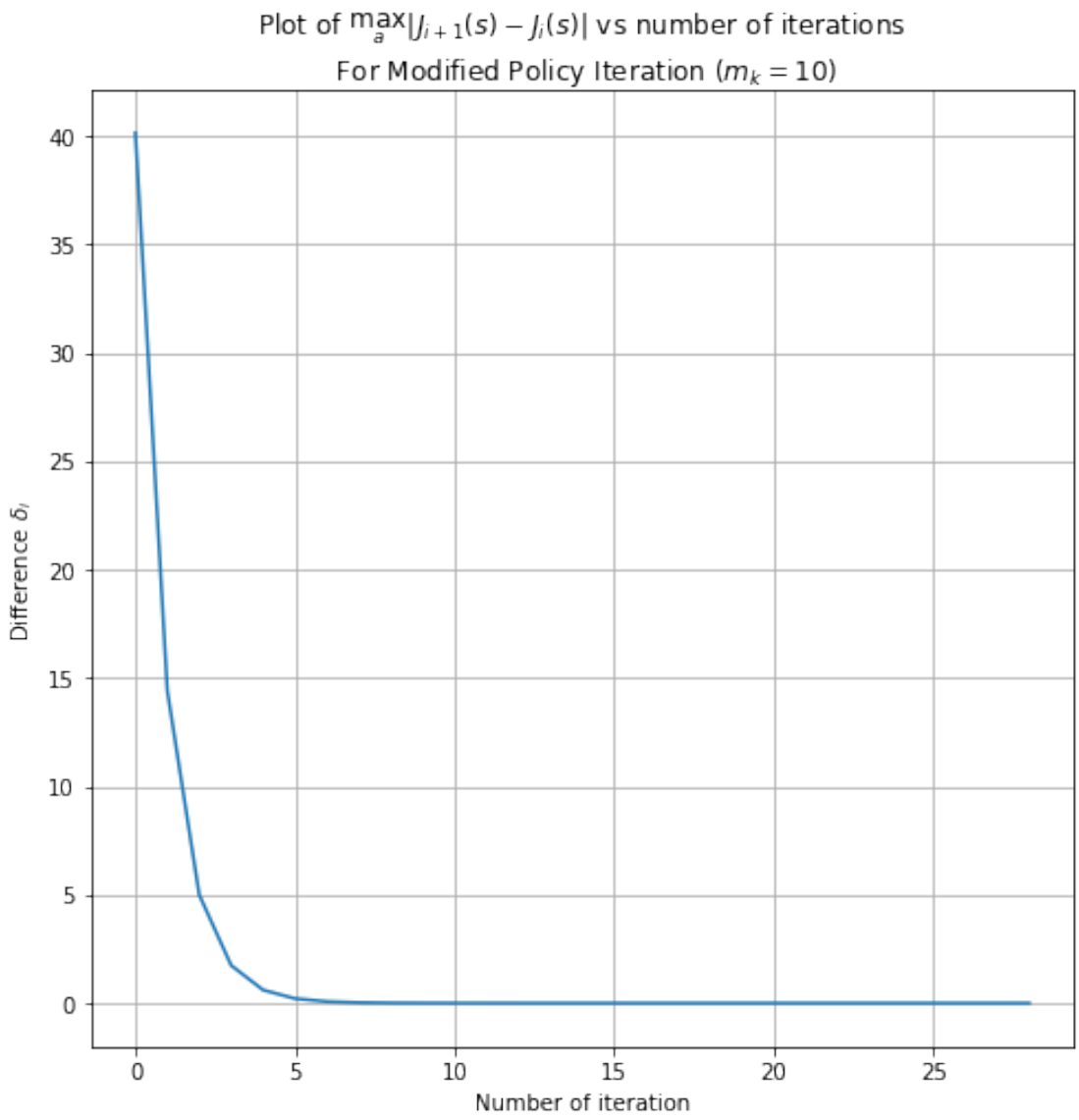
This shows that choosing $m_k = 10$ is better than $m_k = 5$ as we converge faster. This happens because we get a better approximate value of J in each iteration when we apply the T_π operator m_k times and thus leading to a better policy improvement step rather than doing the policy improvement step after the application of T_π operator.

Also note that the optimal costs obtained with modified policy iteration almost match with the optimal costs in the table given before for $\beta = 0.9$




```
Optimal cost with Modified Policy iteration m_k=5 is:
[ 121.65345207  135.30625647  122.83688402]

Optimal actions with Modified Policy iteration m_k=5 is:
[2 2 2]
```




```
Optimal cost with Modified Policy iteration m_k=10 is:
[ 121.65347112  135.30627552  122.83690308]

 Optimal actions with Modified Policy iteration m_k=10 is:
[2 2 2]
```

2.2.1 Value Iteration

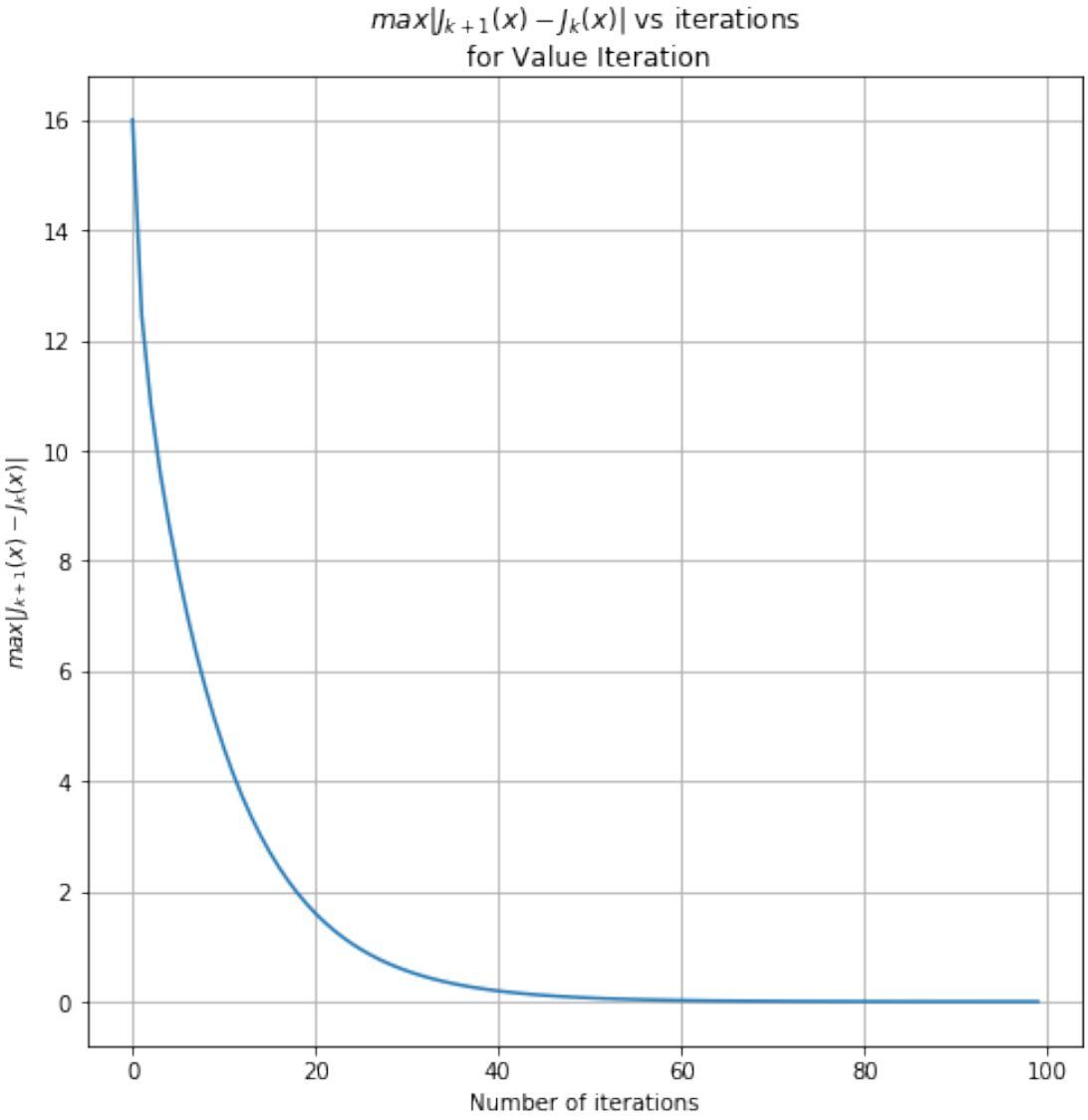
We get the optimal cost of
A : 121.64997255

B : 135.3027769
C : 122.8334045
and optimal actions
A:2
B:2
C:2

Also the plot of $\delta_i = \max_a |J_{i+1}(s) - J_i(s)|$ vs number of iterations For Value Policy Iteration shows that the algorithm converges ($\delta_i < 0.001$) after 60 iterations.

Also note that the optimal costs obtained with value iteration almost match with the optimal costs in the table given before for $\beta = 0.9$

```
##################################################
Starting Value Iteration
```



```
The optimal action obtained from Value iteration after 100 iterations is:
 [2 2 2]
The optimal value obtained from Value iteration after 100 iterations is:
 [ 121.64997255  135.30277695  122.8334045 ]
```

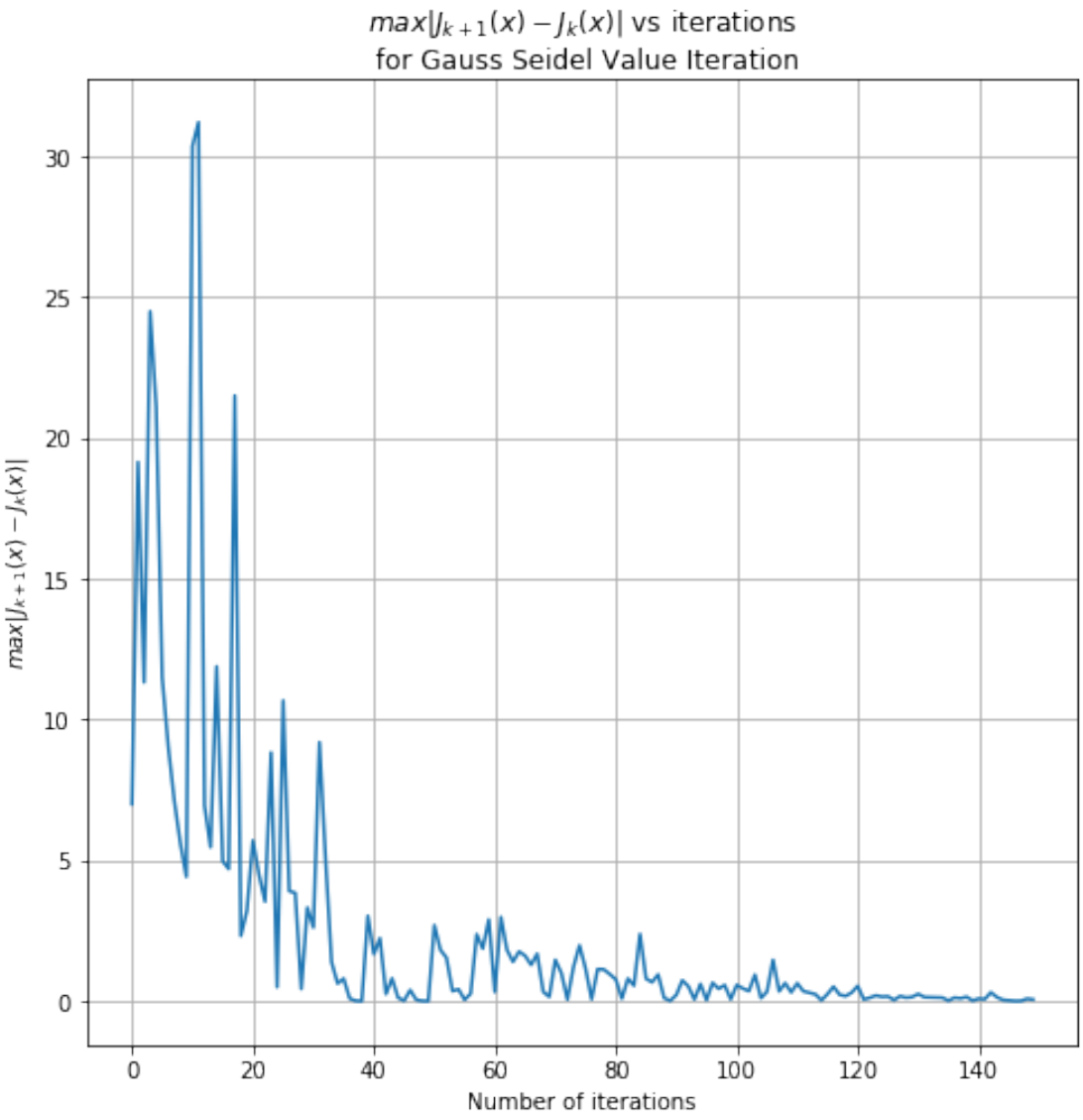
2.2.2 Gauss Seidel Value Iteration

We get the optimal cost of
A : 120.6536124
B : 134.15900833
C : 121.83416938
and optimal actions
A:2
B:2
C:2

Also the plot of $\delta_i = \max_a |J_{i+1}(s) - J_i(s)|$ vs number of iterations For Gauss Seidel Value Policy Iteration shows that the algorithm converges ($\delta_i < 0.001$) after around 60 iterations. The jitter is due to the stochastic nature of the algorithm for choosing the states.

Also note that the optimal costs obtained with Gauss Seidel value iteration almost match with the optimal costs in the table given before for $\beta = 0.9$

```
##################################################
Starting Gauss Seidel Value Iteration
```




```
The optimal action obtained from Value iteration after 100 iterations is:
 [2 2 2]
The optimal value obtained from Value iteration after 100 iterations is:
 [ 121.08398884  134.6711394   122.21113685]
```

2.3 References

1:Class Notes
2:DPOC Book Vol 2