

Playing with the time limit and heuristics

All the raw program output can be found in the *Result-Q5.txt* file.

In the experiments, for comparing the performance of different lower bound heuristics function, we used different time limit on Both **RockSample_4_4** environment and **TagVoid** environment. The time limits and the number of runs to get the average rewards we used are as the following and the precision is 0.1:

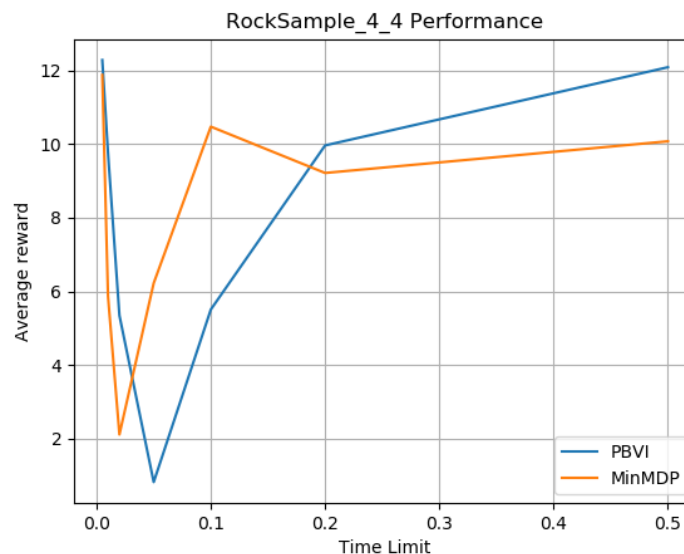
- 5 ms 500 runs
- 10 ms 500 runs
- 20 ms 500 runs
- 50 ms 500 runs
- 100 ms 500 runs
- 200 ms 200 runs
- 500 ms 100 runs

RockSample_4_4

The performance we get with **RockSample_4_4** environment is showing as following:

<i>Time Limit</i>	<i>Average Reward with MinMDP</i>	<i>Average Reward with PBVI</i>
5 ms	11.878012663851935	12.283170287115995
10 ms	5.872868159021878	9.765582201551045
20 ms	2.110595001754537	5.345782413655991
50 ms	6.222406227262964	0.8178194938489292
100 ms	10.473328494859768	5.506114454052732
200 ms	9.215206573303329	9.963284282849001
500 ms	10.07580586386141	12.087644319844621

And the plots of the results are:

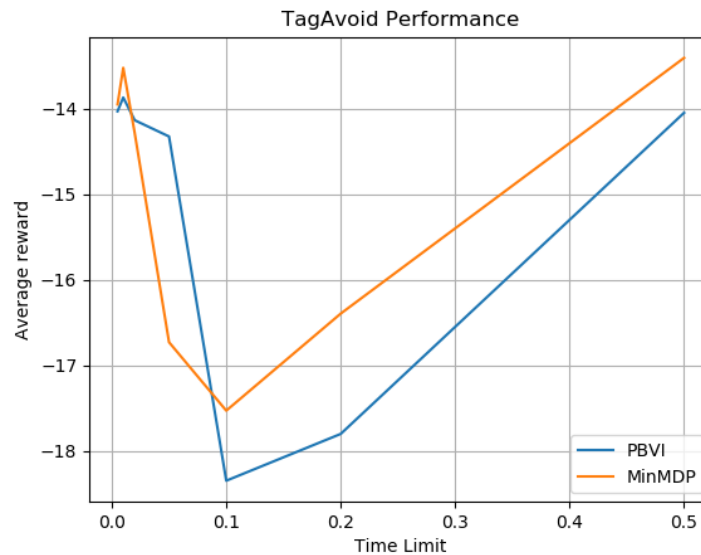


TagAvoid

The performance we get with **TagVoid** environment is showing as following:

Time Limit	Average Reward with MinMDP	Average Reward with PBVI
5 ms	-13.948524759075422	-14.029021829787943
10 ms	-13.5208698315361	-13.868015110265116
20 ms	-14.281603722569985	-14.132935130896012
50 ms	-16.728418354939574	-14.323408076377861
100 ms	-17.528815208911357	-18.34807112465858
200 ms	-16.394293564823176	-17.80084036399528
500 ms	-13.404411914000168	-14.046217864384616

And the plots of the results are:



Analyze

From the results of experiments on both environments, we can find that:

- When the time is very small ($\leq 20\text{ms}$), models that use PBVI as the lower bound will performance better, I think the reason is:
 - During the small mount of time, the POMDP solver cannot expand enough nodes in the And-Or Tree, and in this case, since PBVI already have some policy been pre-made, the differentiate of the upper bound and lower bound might smaller than MinMDP model. And this cause the PBVI model performance better.
- When the time is small but not too small (20ms ~ 50ms in **RockSample_4_4** environment, and 20ms ~ 100ms in **TagAvoid** environment), both models' performance going down and sometime MinMDP is better and other time PBVI will better. I think the reason is:
 - At this time, the POMDP solver can expand some nodes in the And-Or Tree, but still enough, and this cause the solver didn't know what it should to do.
- When the time is a little bit larger (50ms ~ 100ms in **RockSample_4_4** environment, and 100ms ~ 500ms in **TagAvoid** environment), both models's performance going up and become better and

better. And during this time the model with PBVI as lower bound will have better performance. The Reason I think should be:

- With the more time give the POMDP solver, the nodes in the And-Or Tree can be expanded more and more, that can cause the solver finds some path to the better outcome statements.
- When the models have even more time ($\geq 100\text{ms}$ in **RockSample_4_4** environment). The performance of the model continues going up, and the solver model with MinMDP as lower bound will out performance than model with PBVI as lower bound. I think the reason is:
 - Since the POMDP solver have lot of time to expanded the And-Or Tree, and this can let it find the goal states during the first few round of consideration, and not only that, it can even find some new policies which better than the pre-selected and stored in PBVI. That why the MinMDP is better than PBVI at this time.
 - In **TagAvoid** environment, we didn't have the data for it, but I think it should be similar as the **RockSample_4_4** environment and the MinMDP will be better than PBVI if we give more time to the POMDP solver.