1 Experiments

In this section, we report the experimental result of our implementation of Algorithm 1 and 2 presented in previous sections, in the following called `pac-rdp` and `pac-rdp-simple`, respectively. As baseline, we consider vanilla Q-Learning (`q-learning`), trained with ϵ-greedy policy with $\epsilon = 0.1$ and learning rate $\alpha = 0.1$. For Algorithm 2, instead of running `AdaCT` every episode, we run it every 500 episodes; This simplifies the benchmarking, because the workload for each episode would have been too high, without much benefits.

In all the experiments, we run the greddy policy for 50 episodes, with the currently learned model, every 500 training episodes, and collect the average reward obtained across the test episodes. Each experiment is run 8 times , and the plots show the average reward and the standard deviation of the optimal policy during the training, as explained above.

In the final version, the number of runs will be increased

1.1 Rotating MAB

The configurations for this experiments are reported in Table 1.

Parameter	Description	Value
n	Number of arms	2
γ	Discount factor	0.99
ϵ	Accuracy	0.05
δ	Condifidence	0.05
r	Success reward	1
(p_1, p_2)	Success probabilities	$(0.9, 0.2)$
N	Number of episodes	50000
M	Maximum number of steps per episode	100
$\ell_{\max}$	Maximum state upperbound for `pac-rdp`	10
$\hat{D}$	Upperbound on depth for `pac-rdp-simple`	10

Table 1: Parameters for Rotating MAB experiment.

The results are shown in Figure 1. The last model learned as PDFA is shown in Figure 2.

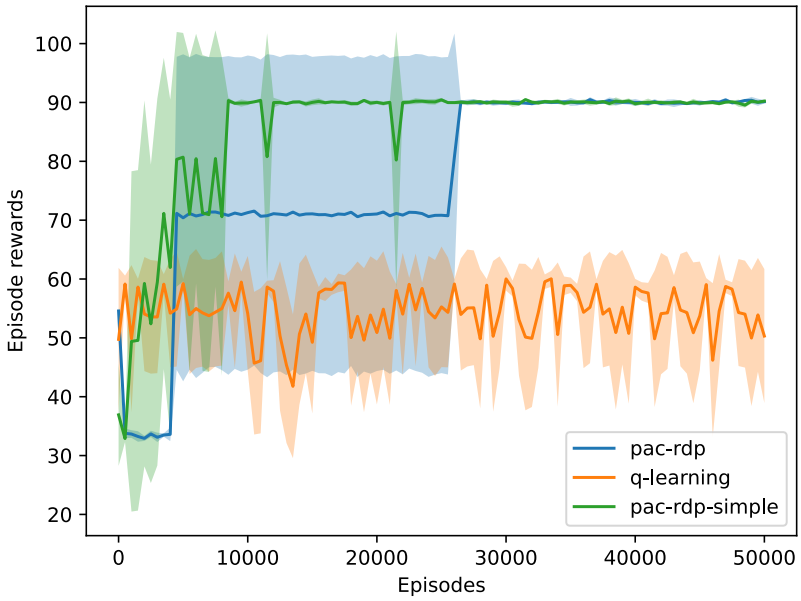

Figure 1: Average episode rewards of the greedy policy for Rotating MAB. The optimal average reward is $M \cdot p_{\max} \cdot r = 0.9$, that is, when the agent always knows the current state of the shifts and the most rewarding arm to pull.

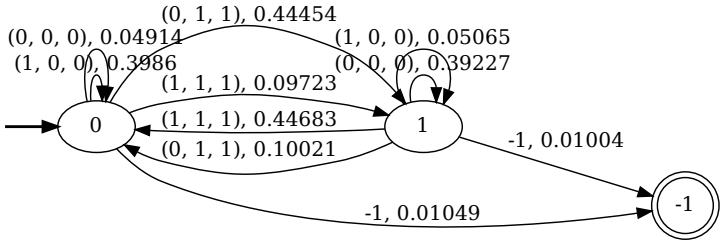

Figure 2: The PDFA for Rotating MAB (p_1, p_2). The transition is made only when a reward is observed (i.e. the middle number of the triple (a, r, s).)

1.2 Cheat MAB

The configurations for this experiments are reported in Table 2.

The results are shown in Figure 3. The last model learned as PDFA is shown

Parameter	Description	Value
n	Number of arms	2
γ	Discount factor	0.99
ϵ	Accuracy	0.05
δ	Condifidence	0.05
r	Success reward	1
P	Pattern to complete	$[0, 0, 1]$
N	Number of episodes	50000
M	Maximum number of steps per episode	100
$\ell_{\max}$	Maximum state upperbound for `pac-rdp`	10
$\hat{D}$	Upperbound on depth for `pac-rdp-simple`	10

Table 2: Parameters for Cheat MAB experiment.

in Figure 4.

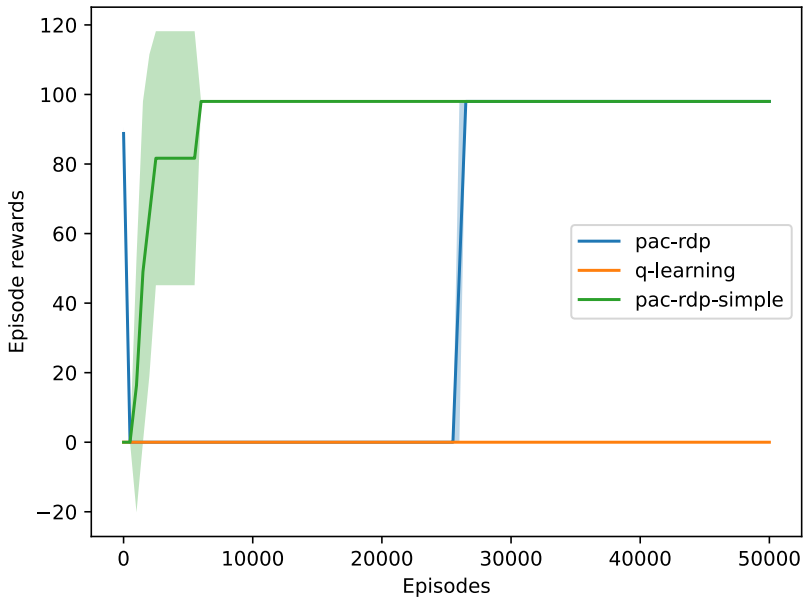

Figure 3: Average episode rewards of the greedy policy for Cheat MAB. The optimal average reward is $(M - length(P) + 1) \cdot r = 98.0$, that is, when the agent completes the pattern as soon as possible.

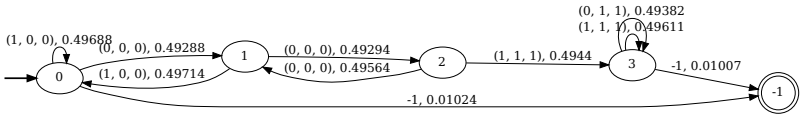

Figure 4: The PDFA for Cheat MAB. The transition is made only when the first number, corresponding to the chosen action, is the right action for the pattern. In this case, the pattern is: two times the action 0, and one time the action 1. After the pattern is completed, reward is always 1, regardless of the actual action.

1.3 Malfunction MAB

The configurations for this experiments are reported in Table 3.

Parameter	Description	Value
n	Number of arms	2
γ	Discount factor	0.99
ϵ	Accuracy	0.05
δ	Condifidence	0.05
r	Success reward	1
(p_1, p_2)	Success probabilities	$(0.2, 0.8)$
k	No. times to break an arm	2
N	Number of episodes	75000
M	Maximum number of steps per episode	100
$\ell_{\max}$	Maximum state upperbound for `pac-rdp`	10
$\hat{D}$	Upperbound on depth for `pac-rdp-simple`	10

Table 3: Parameters for Malfunction MAB experiment.

The results are shown in Figure 5. The last model learned as PDFA is shown in Figure 6.

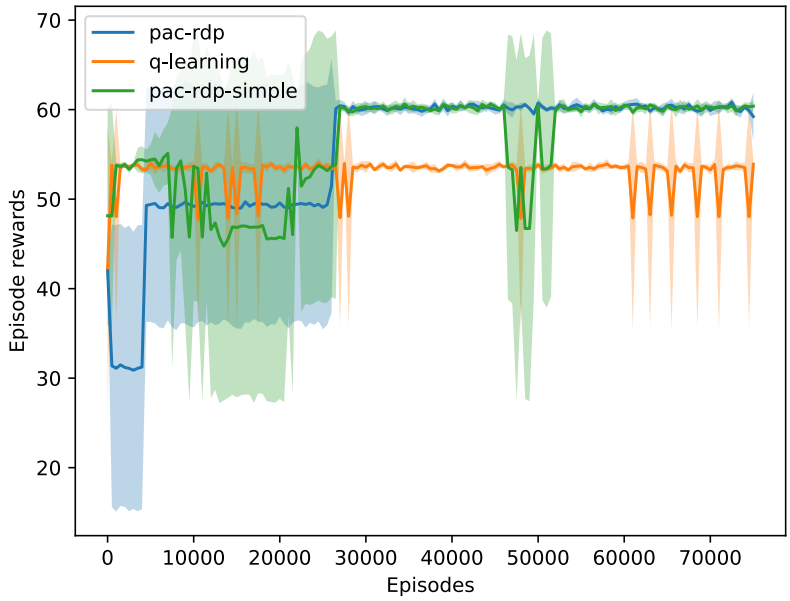

Figure 5: Average episode rewards of the greedy policy for Malfunction MAB. The optimal average reward is the one obtained by pulling the best arm when it is working, and the second best arm in case the first is not working. More precisely, in this case, the optimal average reward is: $Mr(p_1(\frac{k}{k+1}) + p_2(\frac{1}{k+1})) = 60.0$, i.e. the weighted average between the first best (malfunctioning) arm and the second best (working) arm.

1.4 Driving Agent

The configurations for this experiments are reported in Table 4.

Parameter	Description	Value
γ	Discount factor	0.99
ϵ	Accuracy	0.05
δ	Condifidence	0.05
N	Number of episodes	50000
M	Maximum number of steps per episode	100
$\ell_{\max}$	Maximum state upperbound for `pac-rdp`	10
$\hat{D}$	Upperbound on depth for `pac-rdp-simple`	10

Table 4: Parameters for Driving Agent experiment.

The results are shown in Figure 7. The last model learned as PDFA is shown

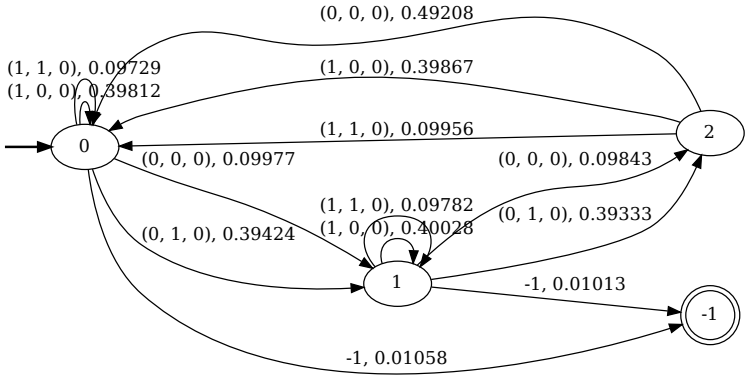

Figure 6: The PDFA for Malfunction MAB. The PDFA reduces to a counter up to k, which increases only when the first arm is pulled. After being in state k, whatever is the next action, the episode goes to the initial state.

in Figure 8.

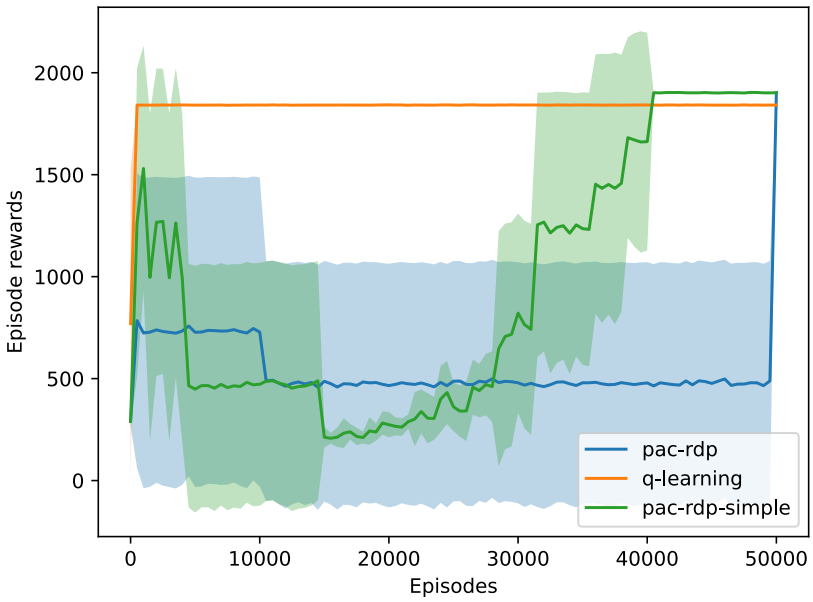

Figure 7: Average episode rewards of the greedy policy for Driving Agent. The optimal average reward is the one obtained by driving slowly when it has been rainy and not sunny since then, and driving normally elsewhere. Q-Learning only learns to drive slowly when not sunny and normally when sunny; however, this is suboptimal, because when cloudy you may drive normally if it has been sunny and not rainy recently.

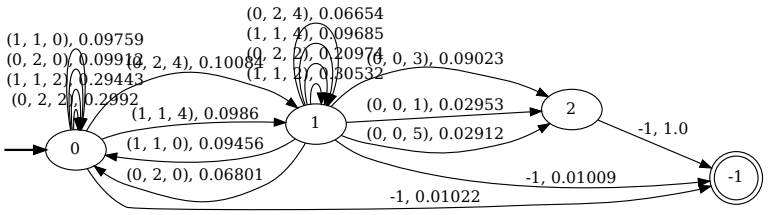

Figure 8: The PDFA for the Driving Agent environment. Action with index 0 is "driving normally", whereas index 1 is "driving slowly". The indices for the rewards are 0 for $r = 0$, 1 for $r = 18$ and 2 for $r = 20$. The indices for the observations S are: even numbers 0, 2 and 4 for, respectively, *sunny*, *cloudy* and *rainy* with no accident; odd numbers 1, 3 and 5 in case an accident is happened. State 0 represents the condition when it has not been rainy recently. When in state 1, it means it has been rainy and not sunny. State 2 means an accident is happened.