Internship report:
Posterior Sampling Reinforcement Learning for olfactory search

Irene Brugnara

August 2021

1 Introduction

This document summarises the work carried out for the curricular DSSC internship by Irene Brugnara under the guidance of prof. Antonio Celani and the advice of Emanuele Panizon.

The project consists in the application of Posterior Sampling Reinforcement Learning to an olfactory search task, comparing with the well-known algorithm of Infotaxis, as well as other heuristic methods based on MDPs (Action Voting and QMDP).

The work comprises implementation of the search algorithms, real-time animations of their trajectories, systematic measurement of performance and statistical analysis of the distribution of search times.

The code is written in Python and optimised with Numpy in order to scale in the grid dimension. The code in hosted on `http://github.com/IreneBrugnara/PSRL-olfactory-search`.

2 Problem statement

The search environment is a rectangular grid of cells representing the positions that the searcher can take (grid-world). At each time step the searcher can move to one of the adjacent cells (up, down, left or right) or stay still. If the searcher is on the right boundary of the grid and applies action right, it stays still (same for the other boundaries). The source of odour is placed in an unknown cell of the grid. The probability of odour detection depends on the relative position of the agent with respect to the source.

The problem of olfactory search can be stated as a partially observable Markov Decision Process: the discrete state space is the set of grid cells; the actions are the five possible moves and the transitions are deterministic; there is a single reward placed in the source cell s^* which is a terminal state; the MDP is infinite-horizon with a discount factor γ. The agent keeps a probability distribution (belief) over the possible source positions and updates it at each time step in a bayesian framework by incorporating the evidence of each new action-observation pair.

3 Observation model

The model of odour detections in atmosphere is based on [1]. The probability of making an observation depends on the angle with respect to the wind direction. The probability of detection at downwind distance x and crosswind distance y is

$$p(x, y) = \cosh^{-2}\left(\frac{Uy}{vx}\right)$$

where U is the mean wind velocity (along x) and v is the amplitude of the turbulent wind component. This is a rough model for the cone in which detections are likely. The parameter v/U represents the cone aperture angle and was set to $v/U = 0.2$.

It is assumed that no observations are possible behind the source (this is a simplification). Thus the likelihood of observing $y \in \{0, 1\}$ in state s having coordinates x, y with respect to the source position is

$$l(y|s) = \begin{cases} Bernoulli(p(x, y)) & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

In addition, the belief b in the state just visited S_t is always put to zero because the source is a terminal state: $b(S_t) = 0$.

4 Setting

The initial prior is chosen to be a uniform distribution over the state space. The search starts only after the first observation is made (without updating the belief before that).

Figure 2 shows that, in the first steps of the search, the belief has two conic regions where the probability is significantly non-zero. The initial state should be chosen to be in the middle of the grid horizontally, so that the problem is perfectly symmetric; otherwise, for example, Thompson Sampling would initially sample more often from the side of the grid containing more states, and this is an advantage if the source is on this side, which introduces a bias.

To measure the performance of the given algorithms, a number of numerical simulations was run under the conditions listed below. Afterwards, some statistical analysis was conducted, taking the average and median of the search time, as well as looking at the whole pdf of the search time.

grid dimension	201 x 151
initial state	(185, 75)
source position	(60, 115)
number of runs	3000
maximum number of iterations	2000
parameter of observation model	0.2

In order to limit the computational burden, a maximum number of iterations is set, after which the search terminates even when the source has not been found yet.

The origin of coordinates is located in the upper-left corner of the grid. The minimum search time possible, i.e. the manhattan distance between initial state and source position, is 165. Given the above setting, the probability of detection in the initial state is ≈ 0.15. Figure 1 shows the probability of detection in each state.

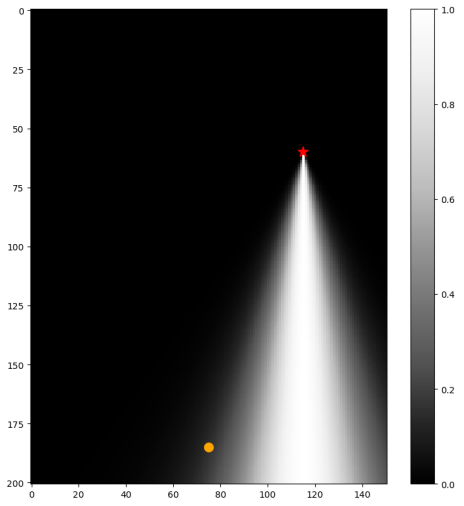


Figure 1: heatmap of the probability of detection (the orange dot is the initial state, the red star is the source)

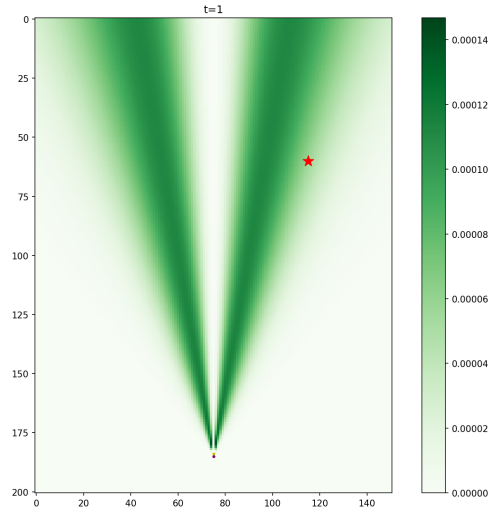


Figure 2: heatmap of the belief at the beginning of the search

5 Algorithms: first part

5.1 Most Likely State

The most straightforward heuristic for choosing actions when solving POMDPs is to select the state which is most likely according to the current belief b and pick the action that is optimal for this state. This is called a greedy policy (or Most Likely State, MLS [2]). In our case of the gridworld, this means to take a step towards the state with maximum probability (the "estimated target"), since the optimal policy for an MDP is the shortest path to the source.

If more than one state has maximum probability, one state is randomly selected between them. If two actions are both optimal, one is randomly chosen.

$$\pi_{MLS}(b) = \pi^*_{MDP}(\operatorname*{argmax}_s b(s))$$

5.2 Thompson Sampling

The greedy policy is sub-optimal because it does not account for uncertainty in the source position. One way to enhance exploration is to introduce some randomisation via Thompson Sampling (TS).

Thompson Sampling randomly draws a state from the current belief and applies the action that is optimal according to it.

$$\hat{s} \sim b(s)$$
$$\pi_{TS}(b) = \pi^*_{MDP}(\hat{s})$$

5.3 Deep exploration

In the spirit of Posterior Sampling Reinforcement Learning [3], instead of re-sampling a new estimate for the source position at each step, we can keep the same hypothesis for a certain number of steps τ. This amounts to preserving the same policy consistently over τ steps. It is called deep exploration and τ is the exploration depth. Deep exploration can be applied to both TS and MLS.

Algorithm 1: Persistent Thompson Sampling

```
initialise b^0(·), τ, S_0, s*, t = 0;
repeat
    sample ŝ ~ b^t(·);
    i = 0;
    while S_t ≠ ŝ and i < τ do
        choose a_t to get one step closer to ŝ;
        apply a_t;
        i = i + 1, t = t + 1;
        reach S_t;
        observe y;
        update b^t(·);
    end
until S_t = s*;
return t;
```

Replacing the third line with

$$\text{choose } \hat{s} = \operatorname*{argmax}_s b^t(s)$$

we obtain a Persistent Greedy algorithm.

5.4 Infotaxis

Infotaxis [4] was implemented as a baseline to compare Thompson Sampling against. Often it happens that the searcher gets stuck nearby the source (a few steps away from the source) for hundreds of iterations. To prevent this, Dual Mode control [2] was applied: when the entropy gets below a certain threshold, the strategy switches from Infotaxis to MLS.

5.5 Description of trajectories

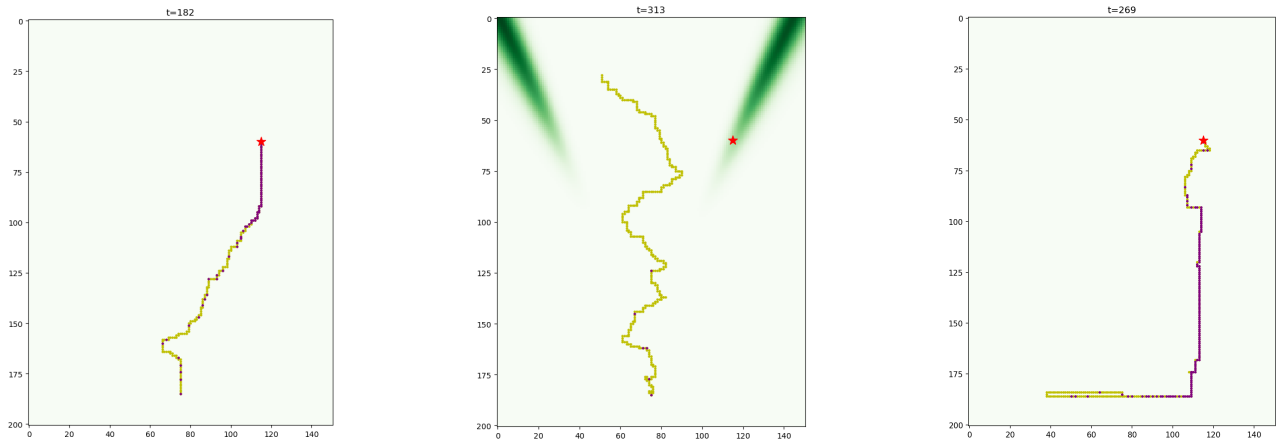
I will describe the typical trajectories observed with the conditions listed in Section 4. Since the environment is perfectly symmetric with respect to the crosswind axis, after the search starts, any search algorithm will turn right or left with equal probability (after going up or down for a certain number of steps, possibly). This initial decision

is particularly important for the search success, as we will see. When describing trajectories in this document, I will always assume that the source is on the right side of the grid for the sake of simplicity.

Deep exploration turns out useful because, in the initial stage of the search, the posterior distribution has two distinct conic regions where it is non-negligible (see Figure 2). If $\tau = 1$, it can happen that at a certain time the sample for the source position comes from one cone so the agent moves one step in this direction, but at the next time step the new sample may come from the other cone, so the agent changes direction, and so on. As a result, the agent gets trapped in this central region of the grid where observations are very unlikely, and it is not able to reach one of the two cones (Figure 3b). Instead, allowing $\tau > 1$ forces the agent to make a decision (of where the source is supposed to be) and to stick with this decision for τ steps. The idea is that if the distance of the agent from the estimated target is less than τ, then it is able to reach it and to discover whether the source is there or not (Figure 3a).

Trajectories of TS and MLS are very similar-looking. When the search with TS or MLS terminates in a reasonable time, this happens because the searcher is able to exclude one of the two cones of probability soon. When the search takes an exceptionally long time, what happens is that the searcher goes back and forth between the two cones, without being able to actually reach one of them, and remains trapped in this central region of the grid where no observations are made.

Infotactic paths exhibit a long horizontal displacement at the beginning of the search. This allows the searcher to exclude one of the two cones of probability. Only when the searcher is sure whether the source is on the right or on the left, it starts to go vertically towards it (Figure 3c). Instead, the greedy strategy points directly to the source and thus is more risky: when the estimation of the source position is good, the search completes in very short time, but when the estimation is poor, the searcher can get lost. This difference in behaviours reflects in the shape of the pdf of the search time, as explained in the next Section.



(a) MLS with $\tau = 15$

(b) MLS with $\tau = 1$ (heatmap of the belief superimposed)

(c) Infotaxis

Figure 3: examples of trajectories (the yellow trace is the sequence of visited states and the purple dots are the observations)

5.6 Performance assessment

The main result is that deep exploration is effective: allowing $\tau > 1$ significantly diminishes search times (in particular for TS). In addition, there seems to be a "sweet spot" for the value of τ: in this case, $\tau = 20$ minimises the average search time, as shown in Figure 4. Second, it turns out that TS does not outperform MLS; instead, it performs slightly worse.

Infotaxis is effective in limiting extremely high search times, but this comes at the expense of a quite high median search time. On the contrary, TS and MLS attain lower search times in the most consistent part of the pdf, but have a great number of events when the search time is exceptionally high (outliers in the pdf).

The pdf of Infotaxis displays two distinct peaks: the one on the right corresponds to trajectories that initially turn to the "wrong" side of the grid, the one on the right corresponds to trajectories that initially turn to the side of the grid containing the source.

Figure 8 shows the benefit of Dual Mode control for Infotaxis.

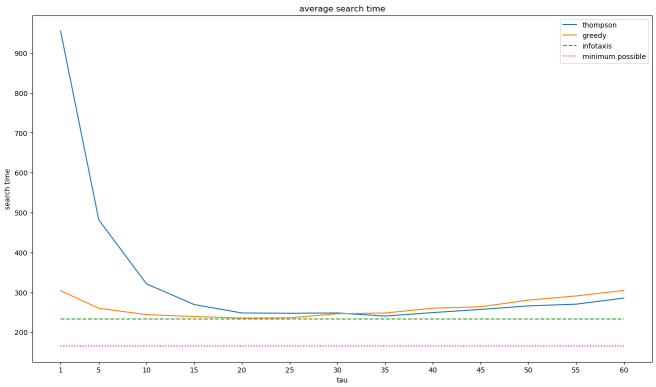


Figure 4: average search time

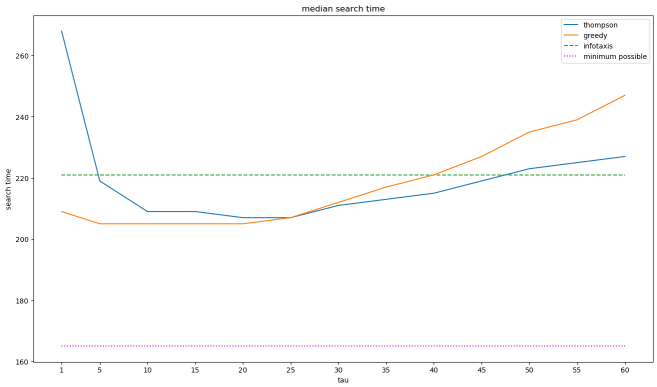


Figure 5: median search time

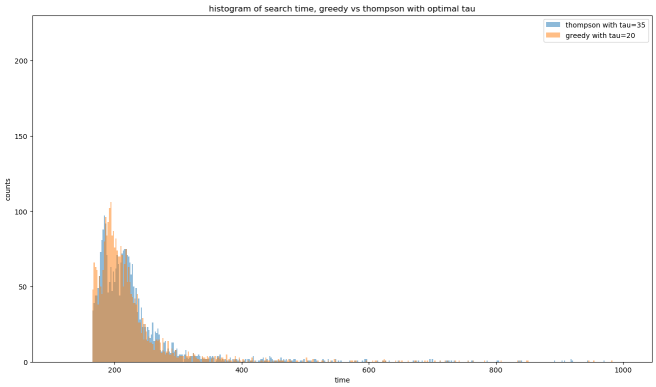


Figure 6: pdf of search time, TS vs MLS

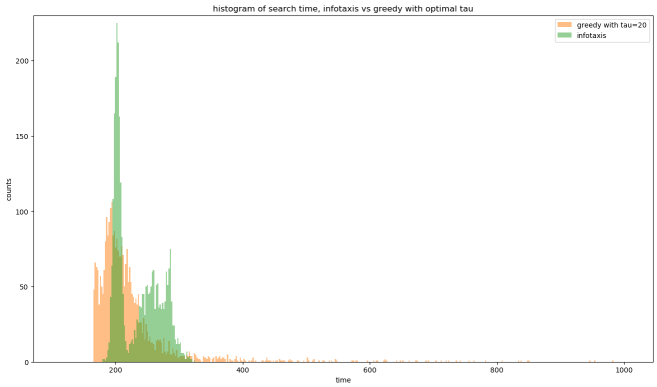


Figure 7: pdf of search time, MLS vs Infotaxis

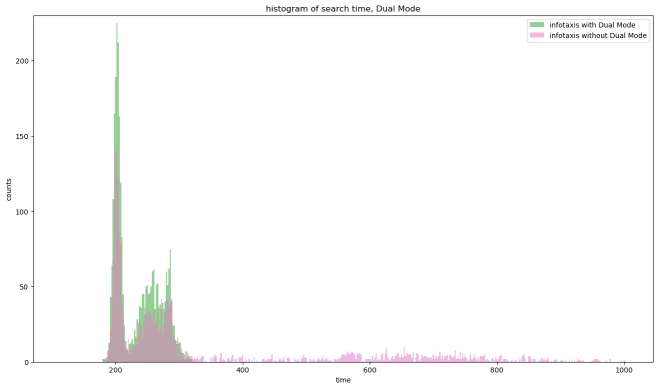


Figure 8: pdf of search time, Infotaxis with and without Dual Mode

6 Algorithms: second part

6.1 MLS with asymmetry

In the MLS algorithm, it can happen that two actions are equivalent. For example, if the estimated source is placed on the top right, the actions "up" and "right" are both optimal. In these cases, one action is always horizontal (right or left) and the other is always vertical (up or down), so instead of selecting one at random, one can decide to select the horizontal action with probability p and the vertical action with probability $1 - p$ ($p = 0.5$ goes back to the previous choice).

6.2 Hybrid greedy+Infotaxis

A drawback of the method in the previous paragraph is that it invokes explicitly the notions of "horizontal" and "vertical" which are justified in the case of wind presence, but not without wind (with radial concentration contours). Another possibility, when two actions are equivalent for MLS, is to select the one which reduces the most the expected entropy of the posterior, i.e. to do Infotaxis between the two equivalent actions.

I used a threshold value equal to 1.5 (tuned by hand), which roughly corresponds to a distance from the source of 2 or 3 steps.

6.3 Adaptive MLS

Instead of keeping τ constant in the MLS algorithm, one can change τ adaptively during the course of the search, based on some property of the belief. For example, suppose that at time t we select as estimate for the source position $\hat{s} = \text{argmax}_s\, b^t(s)$. Then a new estimate will be selected again only when the maximum of the belief at time $t+\tau$, $\max_s b^{t+\tau}(s)$, becomes greater than $c \cdot \max_s b^t(s)$ where c is a fixed coefficient (for the simulations I used $c = 1.5$).

6.4 Results

The number in parenthesis in the legend of Figures 11-13 indicates the number of runs not displayed in the plot (because the histogram is truncated in the tail), which gives an idea of the number of outliers in the distribution of the search time. Increasing the parameter p of the greedy algorithm (Section 6.1) has the effect of reducing the number of outliers (they are completely eliminated with $p = 1$) while keeping the median search time as low as with $p = 0.5$. Greedy adaptive (Section 6.3) fails in this objective. Greedy hybrid (Section 6.2) has very low average and median search times, but the distribution still has a few outliers (the distribution is concentrated around two peaks with very low search time).

The problem with the Adaptive MLS strategy is that, if the searcher gets away from the real source because the estimated target is far from it, no more observations are made, so the belief remains unchanged and the estimated target does not move.

The success of the greedy algorithm with $p > 0.5$ is due to the fact that preferring an horizontal move over a vertical move increases the probability of making an observation, because it makes the agent go more inside the odour plume (see Figure 14). In fact, trying to increase p was motivated by the fact that Infotaxis seems to work so well because of its initial wide horizontal progression (shown in Figure 3c). If one looks at the trajectories, it turns out that the hybrid MLS+Infotaxis algorithm does not always choose the horizontal action over the vertical action when they are equivalent for MLS.

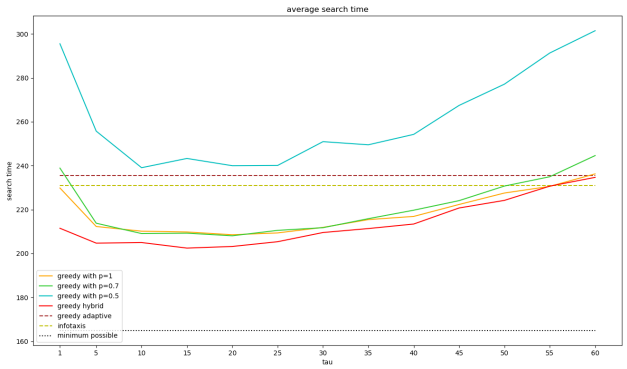


Figure 9: average search time

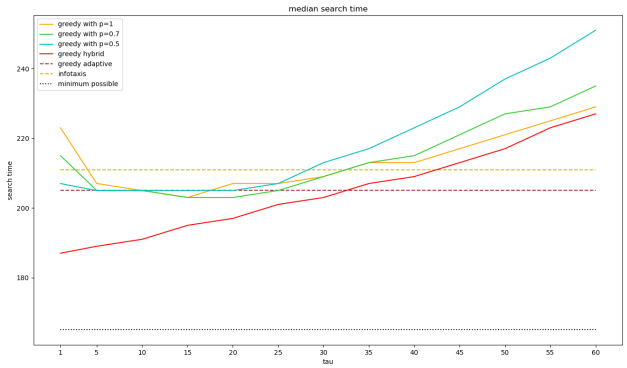


Figure 10: median search time

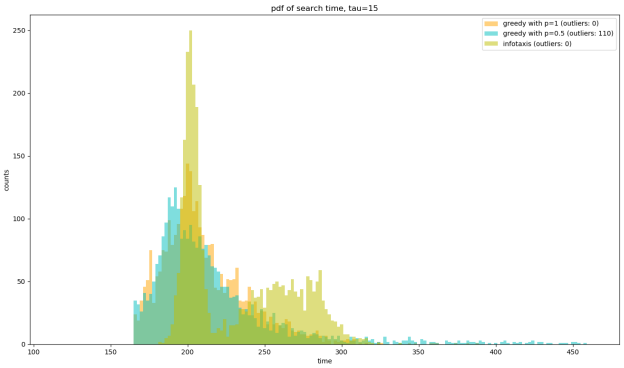


Figure 11: pdf of search time, MLS with $p = 1$

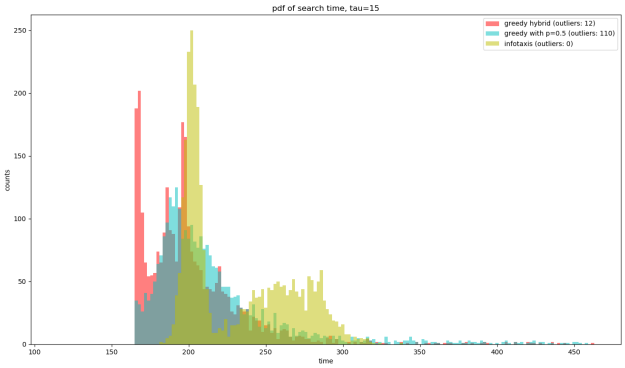


Figure 12: pdf of search time, hybrid MLS+Infotaxis

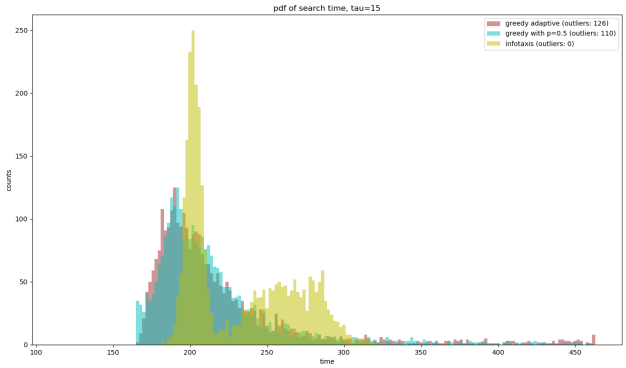


Figure 13: pdf of search time, adaptive MLS

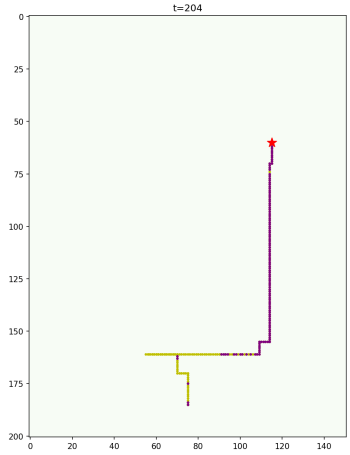


Figure 14: example trajectory of MLS with p=1

7 Algorithms: third part

7.1 Action Voting

Action Voting [2] selects the action which has the highest probability of being the optimal action. The probability that an action is optimal is the sum of the probabilities of the states for which the action is optimal.

$$P_a(b) = \sum_{s \in S} b(s) \cdot \delta(\pi_{MDP}(s), a)$$

$$\pi_{MLS}(b) = \underset{a}{\operatorname{argmax}}\, P_a(b)$$

For the states having two optimal actions, a probability p is assigned to the horizontal action and $1 - p$ to the vertical action (as with MLS). If two actions have equal probability, one is selected at random.

7.2 QMDP

QMDP [2] stems from the consideration that maybe an action is not optimal (so it would not be counted by Action Voting) but still it has a value. QMDP is similar to Action Voting but instead of maximizing the probability that an action is optimal, it maximizes the state-action value function when executing that action:

$$w_a(b) = \sum_{s \in S} b(s) \cdot V_a(s)$$

$$\pi_{MLS}(b) = \underset{a}{\operatorname{argmax}}\, w_a(b)$$

In our case, the state-action value function depends on the manhattan distance between the state reached when executing action a and the source position s: $V_a(s) = \gamma^{d(S_t+a,s)}$ (and $V_a(S_t) = 0$) where S_t is the current state.

7.3 Results

Action Voting with $p = 0.7$ initially swings horizontally back and forth a few times, before going to the right definitively and then quickly reaching the source. With higher p the agent gets stuck as soon as it gets below the source. With lower p ($p = 0.5$), the agent goes straight up for several iteration and then gets stuck.

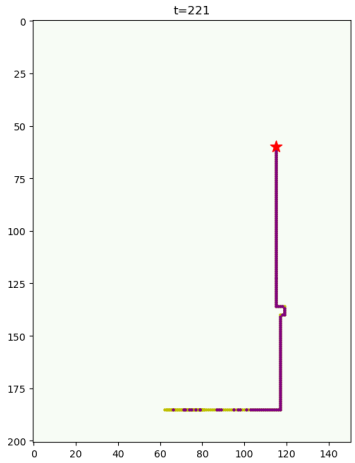


Figure 15: Action Voting with $p = 0.7$

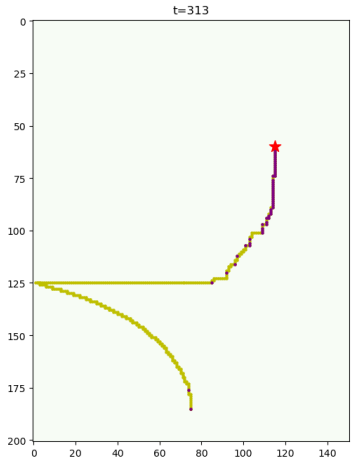


Figure 16: QMDP with $\gamma = 0.8$

With intermediate values of $\gamma \in [0, 1]$ (say $\gamma = 0.8$) in QMDP when the searcher initially turns left, it keeps seeking the source in the left region of the grid, climbing up the cone of probability. At a certain point (often when it reaches the boundary of the grid), it turns right, goes straight right until it enters the real cone and it starts receiving observations, and is able to eventually reach the source. This behaviour is because, for that value of γ, the short-term goal of discovering whether the source is in one of the states in the left side of the grid (which are reachable in just a few steps) is preferable over the long-term goal of traversing the whole grid and getting to the other cone. One could then guess that increasing γ solves this problem. Actually, when γ is higher (say $\gamma = 0.99$), the behaviour is different: the searcher initially goes straight up for several iterations, with the risk of going past the source. If γ is instead too low (say $\gamma = 0.5$), there is a numerical problem: when the searcher gets too distant from the source, the votes $V_a(s)$ go towards zero, reaching the smallest floating-point number that can be represented.

The simulations were conducted with $p = 0.7$ for Action Voting and $\gamma = 0.8$ for QMDP.

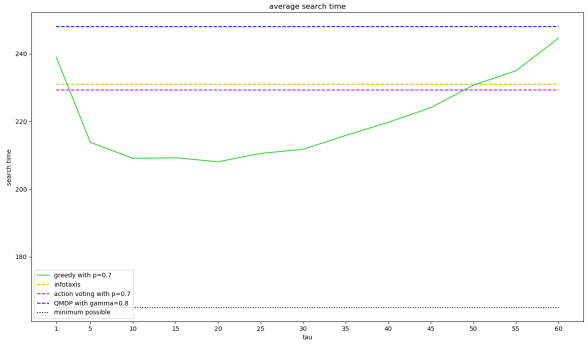


Figure 17: average search time

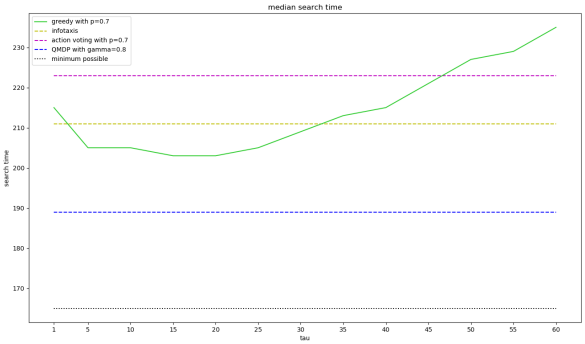


Figure 18: median search time

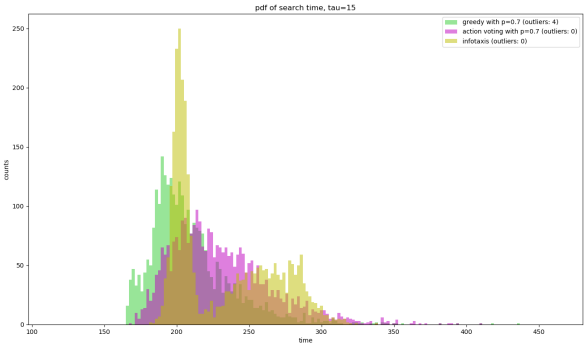


Figure 19: pdf of search time, Action Voting

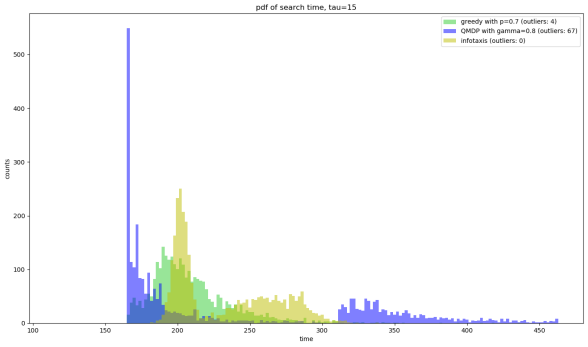


Figure 20: pdf of search time, QMDP

8 Conclusions

To sum up, Thompson Sampling does not overcome the limitations of the greedy strategy in solving this olfactory search problem, while Deep Exploration does.

Thompson introduces some exploration that the greedy strategy lacks but it's random exploration, whereas Infotaxis (and, somehow, Thompson with $p > 0.5$) is exploration in the "right direction" aimed at collecting information for reducing the uncertainty.

Methods like Action Voting or QMDP that are based only on MDPs are not effective enough, because they do not privilege information-seeking actions, like Infotaxis does.

9 Possible further developments

It could be interesting to employ a more realistic model of odour encounters by introducing a threshold of concentration for odour detection, so that the probability of observation no longer depends only on the angle but also on the distance from the source (it decreases exponentially away from the source).

It could be worth studying how the optimal exploration depth for the greedy algorithm scales with the problem size (the grid dimension), in order to fix a good value of τ.

Finally, one could try a persistent version of Action Voting or QMDP, in the same way deep exploration was introduced for TS and MLS. The idea would be to update the belief only every τ steps, so that the policy remains fixed for τ steps.

References

[1] Antonio Celani, Emmanuel Villermaux, and Massimo Vergassola. "Odor landscapes in turbulent environments". In: *Physical Review X* 4.4 (2014), p. 041015.

[2] Joaquín L Fernández et al. "Heuristic anytime approaches to stochastic decision processes". In: *Journal of Heuristics* 12.3 (2006), pp. 181–209.

[3] Ian Osband, Daniel Russo, and Benjamin Van Roy. "(More) efficient reinforcement learning via posterior sampling". In: *arXiv preprint arXiv:1306.0940* (2013).

[4] Massimo Vergassola, Emmanuel Villermaux, and Boris I Shraiman. "'Infotaxis' as a strategy for searching without gradients". In: *Nature* 445.7126 (2007), pp. 406–409.