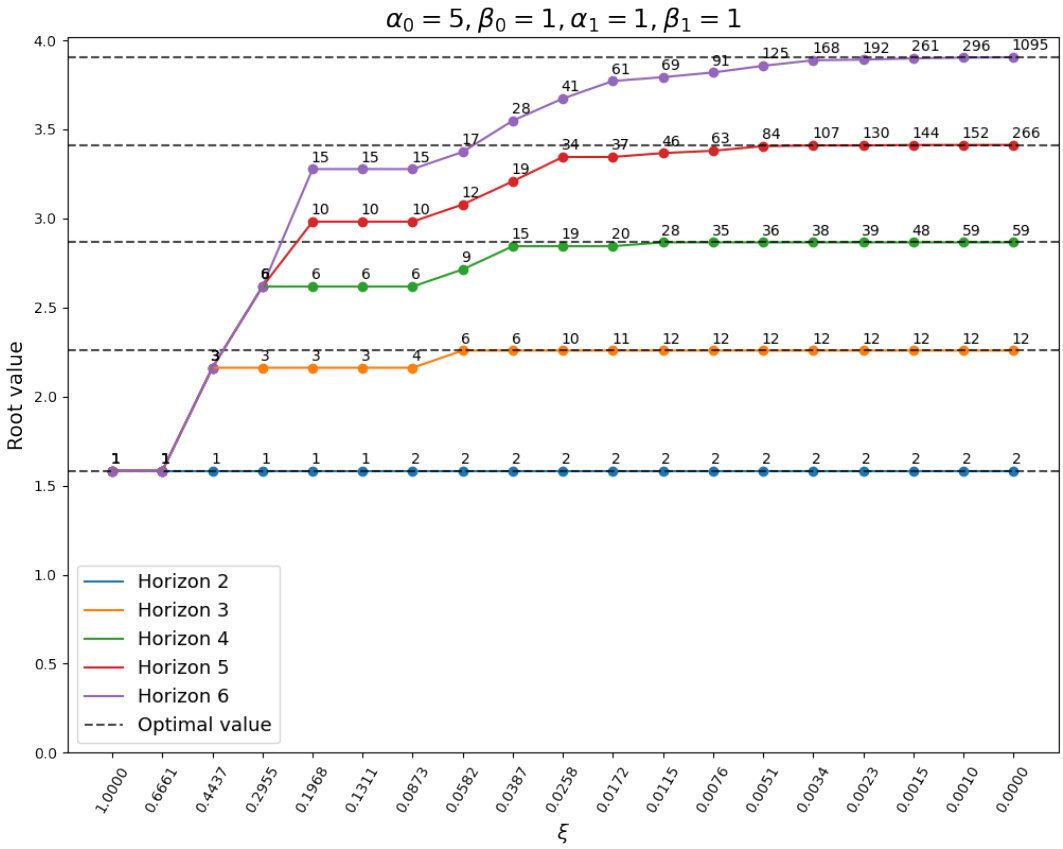
Empirical convergence analysis

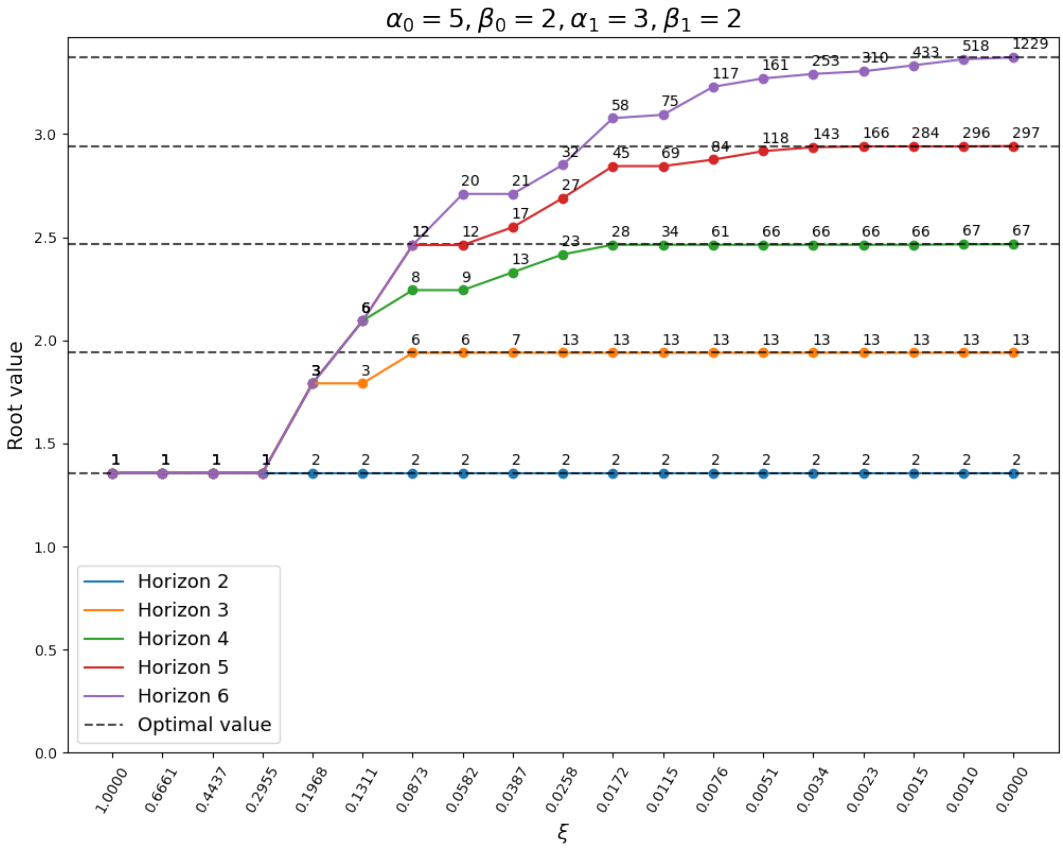
Replay threshold

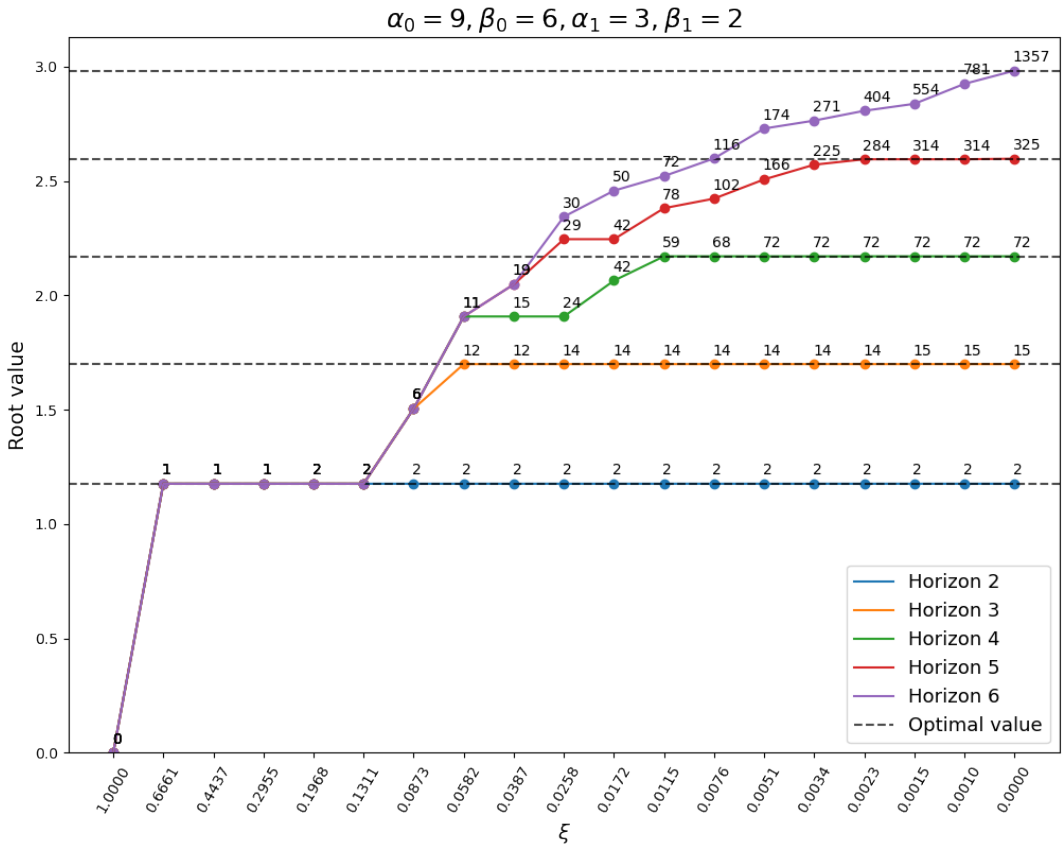
The plots below show how the value of the root state changes with the EVB threshold ξ at multiple horizons and for various prior beliefs. Note that the root Q-values (i.e, the current MF Q-values) were set to 0 everywhere.

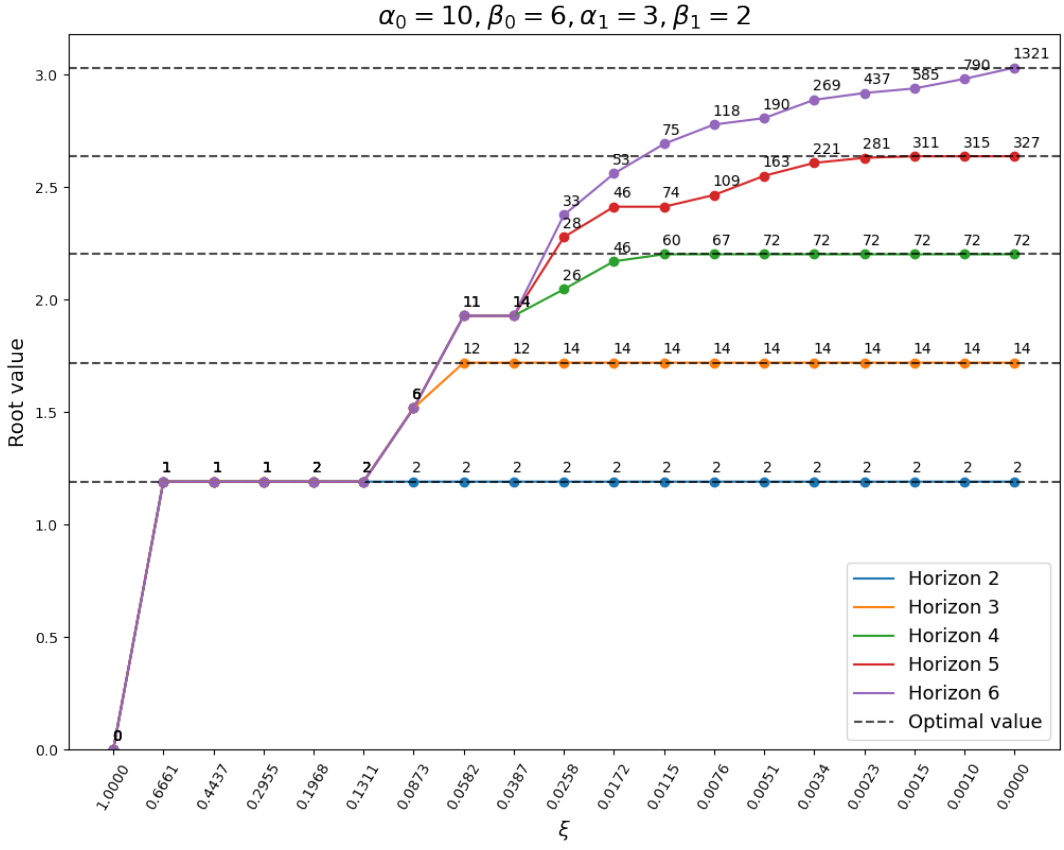
Numbers on top of each data point indicate the number of replays executed at that horizon for that EVB threshold. Titles specify the prior beliefs at the root.

It's nice too see that everything converges to the Bayes-optimal solution when the EVB threshold is set to 0. Another interesting observation is that horizons h need fewer replays to get higher value estimates than the Bayes-optimal value funciton for horizons $h-1$. And, moreover, it seems that the greater the conflict is between the two arms (i.e., in terms of uncertainty and value) – the more replays it takes to converge.


$\alpha_0 = 5, \beta_0 = 1, \alpha_1 = 1, \beta_1 = 1$
Root value
ξ
Horizon 2
Horizon 3
Horizon 4
Horizon 5
Horizon 6
Optimal value


$\alpha_0 = 5, \beta_0 = 2, \alpha_1 = 3, \beta_1 = 2$
Root value
ξ
Horizon 2
Horizon 3
Horizon 4
Horizon 5
Horizon 6
Optimal value


$\alpha_0 = 9, \beta_0 = 6, \alpha_1 = 3, \beta_1 = 2$
Root value
ξ
1.0000 0.6661 0.4437 0.2955 0.1968 0.1311 0.0873 0.0582 0.0387 0.0258 0.0172 0.0115 0.0076 0.0051 0.0034 0.0023 0.0015 0.0010 0.0000
Horizon 2
Horizon 3
Horizon 4
Horizon 5
Horizon 6
Optimal value


$\alpha_0 = 10, \beta_0 = 6, \alpha_1 = 3, \beta_1 = 2$
Root value
ξ
1.0000 0.6661 0.4437 0.2955 0.1968 0.1311 0.0873 0.0582 0.0387 0.0258 0.0172 0.0115 0.0076 0.0051 0.0034 0.0023 0.0015 0.0010 0.0000
Horizon 2
Horizon 3
Horizon 4
Horizon 5
Horizon 6
Optimal value

Number of replays

I also created heatmaps for the different combinations of prior beliefs, where the values illustate how many replays it takes to converge to the true value function.

These plots look rather strange – and it seems like there is some sort of oscillatory nature to those patterns. One issue that came up with these plots is the small visible irregularities which I think are a result of (potentially) numerical errors – but I haven't quite managed to resolve those yet. I will try to fix that.

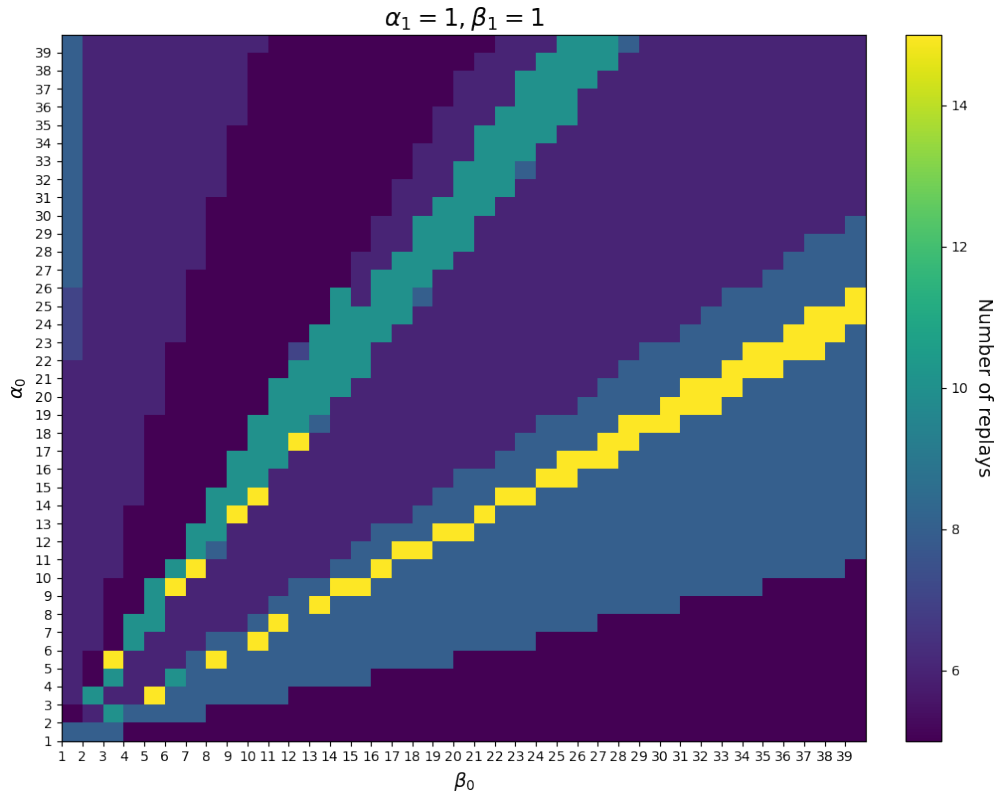


Figure 1: Horizon 2

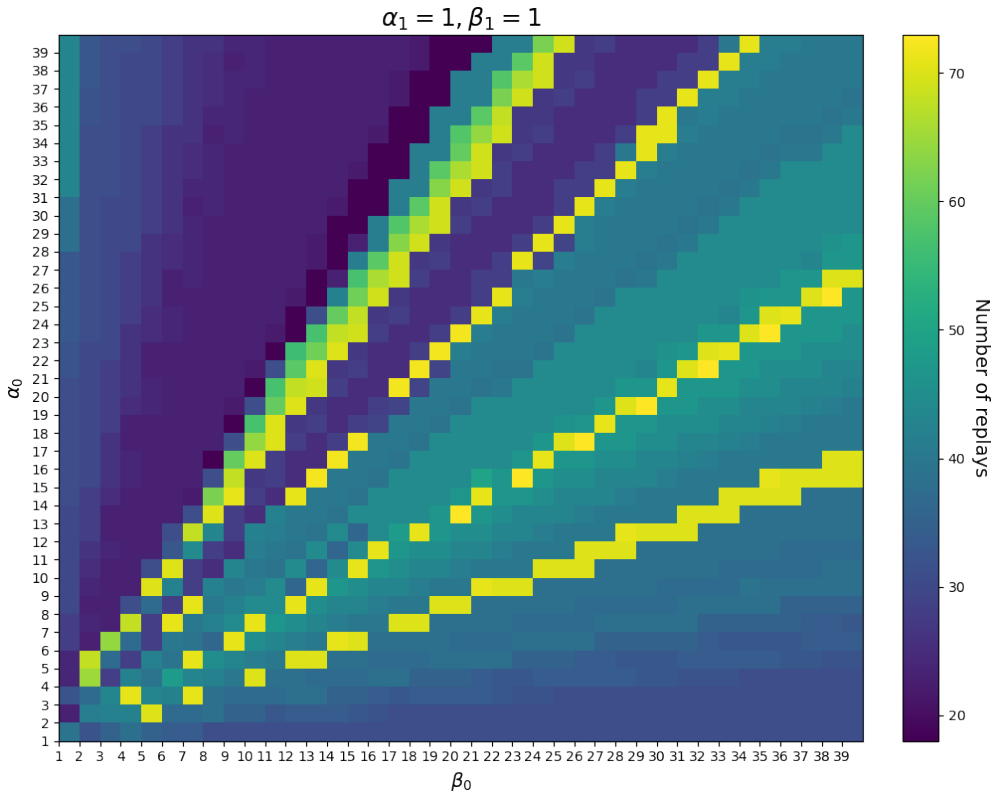


Figure 2: Horizon 3

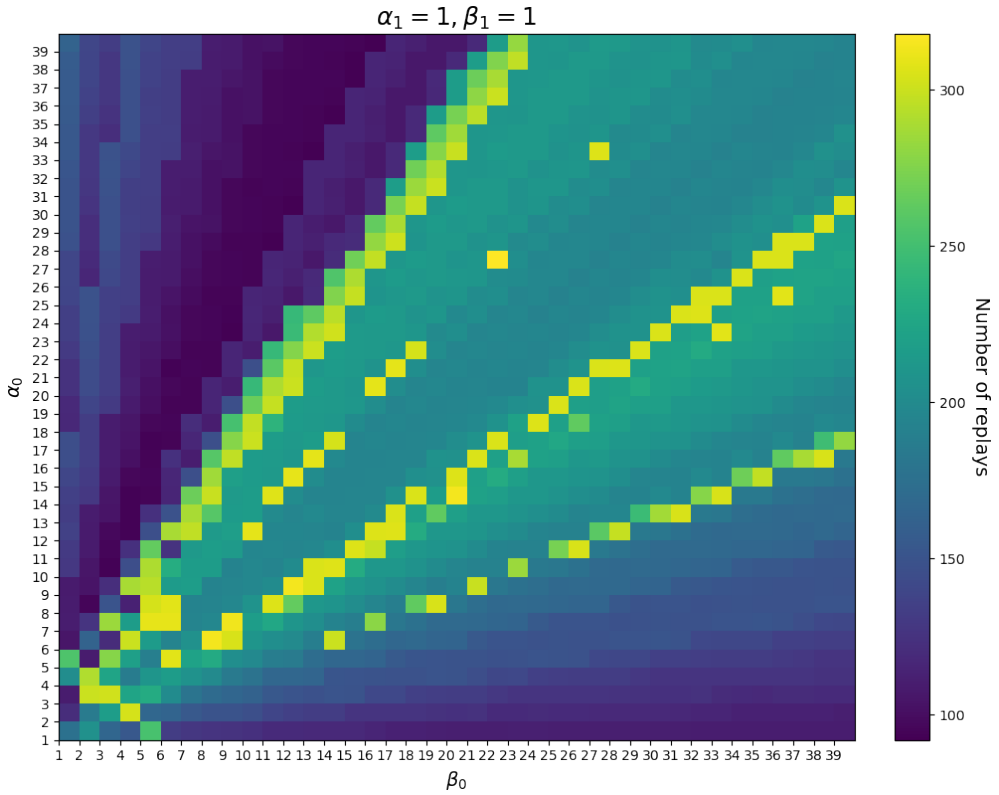


Figure 3: Horizon 4

Maze simulations

I also simulated the maze environment. The prioritisation of each experience was detemined by:

$$EVB(< s, b >, a) = p(b_\rho \rightarrow b, \pi) \times \text{Need}(s \mid b, \pi) \times \text{Gain}(s, a \mid b, \pi)$$

where $p(b_\rho \rightarrow b, \pi)$ is the probability of reaching belief b from the root belief b_ρ under the current policy π, $\text{Need}(s \mid b, \pi)$ is the Need of state s computed with the transition model implied by the belief b and the current policy, and $\text{Gain}(s, a \mid b)$ is the Gain associated with updating the Q-value $Q(< s, b >, a)$ towards the expected value of the next belief, $\mathbb{E}_{b', s' \sim p(b', s' | b, s, a, \pi)}[R(s') + \gamma v(< b', s' >)]$.

The implementation is very similar to that of the bandit case – during planning (replay), the agent generates a planning tree from every state and for every action. The updates can then happen either in distal parts of the trees (only 1 update is allowed at each iteration), or at the root. The value of each node is initialised to the agent's current MF Q-values

There is a strange limitation to this prioritisation scheme: namely, it only considers what happens *locally* to the value function considered for an update. In the bandit case this wasn't so much of an issue – since all updates are indeed local and only affect the updated nodes. In the maze, however, it's different – if one of the root action values is updated, then this also changes the initialised node values in trees that stem from other nearby state-actions (if those are within the reach depending on the horizon).