Question 1

We first note that the LP formulation is akin to

$$\begin{aligned}
&\min_x \sum_{j\in\mathcal{A}_1} c_j x_j + ... + \sum_{j\in\mathcal{A}_m} c_j x_j \\
&\quad s.t. \sum_{j\in\mathcal{A}_1} (\mathbf{e}_1 - \gamma \mathbf{p}_j) x_j + ... + \sum_{j\in\mathcal{A}_m} (\mathbf{e}_m - \gamma \mathbf{p}_j) x_j = \mathbf{e} \\
&\quad x_j \geq 0 \ \forall j
\end{aligned}$$

or in matrix form: if we allow

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad A = C - N, \quad N_{ij} = \gamma p_{ji}, \quad C_{i,j} = \begin{cases} 1 \text{ if } j \in \mathcal{A}_i \\ 0 \text{ otherwise} \end{cases}$$

we have

$$\begin{aligned}
&\min_X c^T X \\
&s.t.\ AX = \mathbf{e},\ \mathbf{X} \geq \mathbf{0}
\end{aligned}$$

We first show that there must be a feasible solution. We consider the alternative system pair. $-A^T y \geq 0,\ e^T y > 0$. This implies that $\sum_i y_i > 0,\ -(y_j - \gamma \sum_k p_{j,k} y_k) \geq 0\ \forall j \in A_i, \forall i$ in other words $e^T y > 0,\ (\gamma p_j^T - e_i^T) y \geq 0\ \forall j \in A_i, \forall i$. We note that the first constraint implies that $\max_i y > 0$. However this would imply that allowing $k = arg\max_i y_i$,

$$(\gamma p_j^T - e_k^T) y \geq 0\ \forall j \in \mathcal{A}_k$$

However as $\sum_i p_{ji} = 1$, we must have that $p_j^T y \leq \max_i y_i$, thus as $\gamma \in (0, 1)$ we would have

$$0 > (\gamma e_k^T - e_k^T) y \geq (\gamma p_j^T - e_k^T) y \geq 0\ \forall j \in \mathcal{A}_k$$

Which is clearly a contradiction. Thus, as the alternative system is infeasible, there must be a feasible solution to the original problem $Ax = b,\ x \geq 0$

Now, we note that the original problem is equivalent to

$$\begin{aligned}
&\min_x \sum_{j\in\mathcal{A}_1} c_j x_j + ... + \sum_{j\in\mathcal{A}_1} c_j x_j \\
&\quad s.t. \sum_{j\in\mathcal{A}_i} x_j - \gamma \sum_{k\in[m]} \sum_{j\in\mathcal{A}_k} p_{ji} x_j = 1\ \&\ x_j \geq 0,\ \forall i \in [m], \forall j
\end{aligned}$$

We note that as $x_j \geq 0$ and $\mathbf{p}_j$ is the state transition probabilities, we must have that $\sum_{j\in\mathcal{A}_i} x_j \geq 1$ or otherwise our second constraint would be violated.

Next, as the problem is feasible so there must be a basic feasible solution, we note that there must be a solution with only m nonzero variables. Because by Caratheodory's theorem if $\exists x$ feasible for $\{x|Ax = b, x \geq 0\}, A \in \mathbb{R}^{m\times n}$, then exists basic feasible solution with m nonzero entries. Therefore, as at least one x_j corresponding to an $\mathcal{A}_i$ must be greater than zero, and there are m $\mathcal{A}_i$'s, we must have a basic optimal solution with exactly one x_i nonzero for each $\mathcal{A}_j$. Moreover, as we previously showed, $\sum_{j\in\mathcal{A}_i} x_j \geq 1\ \forall i$, then there exists an optimal basic solution such that there the basic variables have exactly one variable from each state i, and are bounded below by 1.

WLOG let us denote that x_{jk} as the unique nonzero value for each $\mathcal{A}_j$ (because we can always reorder the actions in the set of action $\mathcal{A}_j$, s.t. action k is the one that is picked from state j). Our constraint set then becomes

$$x_{ik} - \gamma \sum_{j\in[m]} p_{ji}x_{jk} = 1 \ \&\ x_i \geq 0 \quad \forall i \in [m], \forall j$$

This gives us that

$$\sum_i x_{ik} - \gamma \sum_i \sum_j p_{ji}x_{jk} = m \ \Rightarrow \ \sum_i x_{ik} - \gamma \sum_j \sum_i p_{ji}x_{jk} = m \ \Rightarrow$$
$$\sum_i x_{ik} - \gamma \sum_j x_{jk} \sum_i p_{ji} = m \ \Rightarrow \ \sum_i x_{ik} - \gamma \sum_j x_{jk} \cdot 1 = m \ \Rightarrow \ \sum_i x_{ik} = \frac{m}{1-\gamma}.$$

Thus there are exactly m nonzero basic variables and they sum to $\frac{m}{1-\gamma}$.

Question 2

Allowing $U(\cdot)$ an update function that maps y^k to y^{k+1}, i.e. $U(y^k) = y^{k+1}$. We have that

$$\begin{aligned}\left\|y^{k+1} - y^*\right\|_\infty &= \left\|U(y^k) - U(y^*)\right\|_\infty \\ &= \max_i(\min_{j\in\mathcal{A}_i}(c_j + \gamma \mathbf{p_j^T y^k}) - (\min_{\mathbf{j}\in\mathcal{A}_\mathbf{i}} \mathbf{c_j} + \gamma \mathbf{p_j^T y^*}))\end{aligned}$$

consider for any i

$$\begin{aligned}|y_i^{k+1} - y_i^*| &= |\min_{j\in\mathcal{A}_i}(c_j + \gamma \mathbf{p_j^T y^k}) - (\min_{\mathbf{j}\in\mathcal{A}_\mathbf{i}} \mathbf{c_j} + \gamma \mathbf{p_j^T y^*})| \\ &= |-1 * (\max_{j\in\mathcal{A}_i}((-c_j - \gamma \mathbf{p_j^T y^k}) - (\max_{\mathbf{j}\in\mathcal{A}_\mathbf{i}} -\mathbf{c_j} - \gamma \mathbf{p_j^T y^*})))| \\ &= |\max_{j\in\mathcal{A}_i}((-c_j - \gamma \mathbf{p_j^T y^k}) - (\max_{\mathbf{j}\in\mathcal{A}_\mathbf{i}} -\mathbf{c_j} - \gamma \mathbf{p_j^T y^*}))| \\ &\leq |\max_{j\in\mathcal{A}_i}(-c_j - \gamma \mathbf{p_j^T y^k}) - (-\mathbf{c_j} - \gamma \mathbf{p_j^T y^*})|\end{aligned}$$

$$|y_i^{k+1} - y_i^*| = |\max_{j\in\mathcal{A}_i} -\gamma \mathbf{p_j^T y^k} + \gamma \mathbf{p_j^T} \mathbf{y}^*)| = |\max_{\mathbf{j}\in\mathcal{A}_\mathbf{i}} \gamma \mathbf{p_j^T}(\mathbf{y^k} - \mathbf{y}^*)|$$
$$= |\max_{j\in\mathcal{A}_i} \gamma \sum_m p_{jm}(\mathbf{y_m^k} - \mathbf{y_m^*})|$$

now we note that as each $p_{ik} \in [0,1]$ and $\sum_k p_{ik} = 1$

$$|\max_{j\in\mathcal{A}_i} \gamma \sum_m p_{jm}(\mathbf{y_m^k} - \mathbf{y_m^*})| \leq |\max_m \gamma(\mathbf{y_m^*} - \mathbf{y_m^k})| \leq \max_{\mathbf{m}} |\gamma(\mathbf{y_m^*} - \mathbf{y_m^k})|$$
$$= \gamma \left\| \mathbf{y_k^*} - \mathbf{y_k^k} \right\|_\infty = \gamma \left\| \mathbf{y^k} - \mathbf{y}^* \right\|_\infty$$

thus we have that

$$\left\| y^{k+1} - y^* \right\|_\infty = \max_i |y_i^{k+1} - y_i^*| \leq \gamma \left\| \mathbf{y^k} - \mathbf{y}^* \right\|_\infty$$

Question 3

We note first that

$$\min_{j\in\mathcal{A}_i}\{c_j + \gamma \mathbf{p_j^T y}^*\} = y_i^* \ \forall i$$

This implies that if $y \geq y^*$ then we must have that

$$\min_{j\in\mathcal{A}_i}\{c_j + \gamma \mathbf{p_j^T y}\} \geq y_i^* \ \forall i$$

as if $y \geq y^*$ then we must have that $\mathbf{p_j^T y} \geq \mathbf{p_j^T y}^*$ as $\mathbf{p} \geq \mathbf{0}$.

This immediately implies that if $y_0 \geq y^*$, then $y^k \geq y^*$ for all k. Therefore, we now turn our attention to proving that $y_i^{k+1} \leq y_i^k \forall i$.

We will prove this via induction. First, assume that $y_m \geq y_{m+1} \forall m < k$.

We note that

$$y_i^{k+1} = \min_{j\in\mathcal{A}_i}\{c_j + \gamma \mathbf{p_j^T y^k}\} \leq \min_{j\in\mathcal{A}_i}\{c_j + \gamma \mathbf{p_j^T y^{k-1}}\} = y_i^k.$$

Thus, if $y^0 \geq y^*$ and $y^1 \leq y^0$, then

$$y^* \leq y^{k+1} \leq y^k, \ \forall k$$

Now if we instead have that:

$$y_i^0 \leq \min_{j\in\mathcal{A}_i}\{c_j + \gamma \mathbf{p_j^T y^0}\} \ \forall \mathbf{i}$$

We will first show that if $y^k \leq y^*$, then $y^{k+1} \leq y^*$. We note that as $\mathbf{p} \geq \mathbf{0}$, we must have that if $y^k \leq y^*$, then $\mathbf{p_j^T y^k} \leq \mathbf{p_j^T y}^*$. thus as

$$y_i^{k+1} = \min_{j \in \mathcal{A}_i}\{c_j + \gamma \mathbf{p_j^T y^k}\} \leq \min_{\mathbf{j} \in \mathcal{A}_\mathbf{i}}\{\mathbf{c_j} + \gamma \mathbf{p_j^T y^*}\} = \mathbf{y^*}$$

Thus if $y^k \leq y^*$, then $y^{k+1} \leq y^*$. Now, we note that as

$$y_i^0 \leq \min_{j \in \mathcal{A}_i}\{c_j + \gamma \mathbf{p_j^T y^0}\}\ \forall \mathbf{i} \ \ \& \ \ \mathbf{y_i^1} = \min_{\mathbf{j} \in \mathcal{A}_\mathbf{i}}\{\mathbf{c_j} + \gamma \mathbf{p_j^T y^0}\}\ \forall \mathbf{i}$$

we must have that $y^1 \geq y^0$. Thus we can build an inductive argument similar to the previous one. Assume that $y^m \geq y^{m-1}\ \forall m \leq k$.

Now we note that

$$y_i^{k+1} = \min_{j \in \mathcal{A}_i}\{c_j + \gamma \mathbf{p_j^T y^k}\} \geq \min_{\mathbf{j} \in \mathcal{A}_\mathbf{i}}\{\mathbf{c_j} + \gamma \mathbf{p_j^T y^{k-1}}\} = \mathbf{y_i^k}$$

Thus by induction as we have previously shown the base case, we must have that if

$$y_i^0 \leq \min_{j \in \mathcal{A}_i}\{c_j + \gamma \mathbf{p_j^T y^0}\}\ \forall \mathbf{i}$$

then $y_k \geq y^{k-1}$ and as we have shown if $y^k \leq y^*$ then $y^{k+1} \leq y^*$ therefore if

$$y_i^0 \leq \min_{j \in \mathcal{A}_i}\{c_j + \gamma \mathbf{p_j^T y^0}\}\ \forall \mathbf{i} \ \Rightarrow \ \mathbf{y^k} \leq \mathbf{y^{k+1}} \leq \mathbf{y^*}, \ \forall \mathbf{k}.$$

Experiments

Throughout our experiments we maintained 2D Maze environment, while varying a grid world size $\{50 \times 50, 100 \times 100, 200 \times 200\}$. We solve the Bellman equation for the optimal Value Function using Value Iteration (VI) for a state space of size $\{2.5 \cdot 10^3, 10^4, 4 \cdot 10^4\}$ respectively, and randomly pick $\{4, 8, 32\}$ terminal states T. Since it is a random maze, transition to any of the states in T has reward $r \in \{-1, 0, 1\}$, while all other transitions cost 0. We explored various settings, varying the number of possible actions from each state within $|\mathcal{A}| = \{1, 2, 4, 8\}$, because every cell has at most 8 neighbors.

Question 4

Random-K Value Iteration

In contrast to standard VI, at each iteration t Random VI method updates state values only for a randomly selected subset $B^{(t)}$, $|B^{(t)}| = k$. Hence, we expect standard VI to converge faster than Random VI as the reward signal would propagate much faster from the cells neighboring to the terminal ones to the rest of the states and in more coherent way than when sampling a subset of states to update.

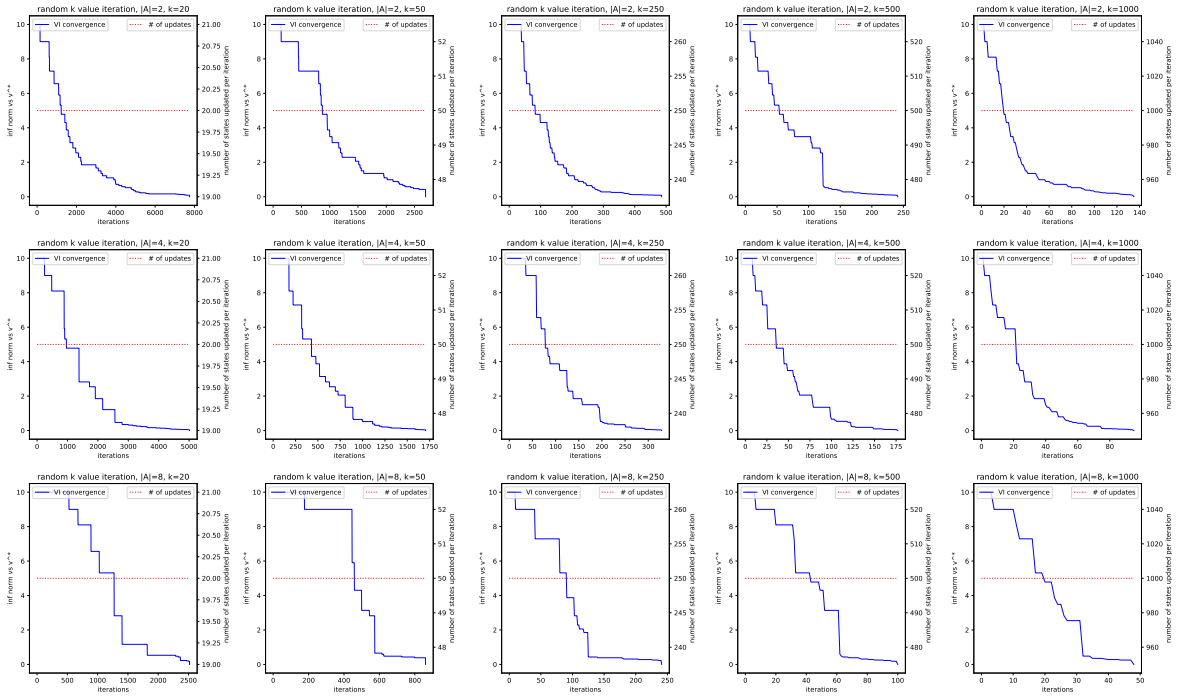

Figure 1: Random VI, dependency on k and $|\mathcal{A}|$. Maze size 50×50.

This intuition is supported by our experiments (see Figure 1 top two rows), e.g when it takes VI to converge in 18 steps, Random VI converges in 850 steps (for $|\mathcal{A}| = 8$, $k = 50 = |B^{(t)}|$). Therefore, when increasing $k \in \{20, 50, 250, 500, 1000\}$, the convergence of Random VI goes from 2500 to 50 (see Figure 1 bottom row).

However, the number of updates per iteration is significantly smaller in Random VI (it equals exactly to k, while in VI it equals to the total state space size). So when in VI we do $20 \cdot 2500 = 5 \cdot 10^4$ unit updates (converges in 20 epochs, $2.5 \cdot 10^3$ updates per epoch), in Random VI we do $8500 \cdot 50 = 4.25 \cdot 10^4$ unit updates.

Influence Tree Value Iteration

We test two variants of influence tree value iteration. In the first, we start by updating all states, i.e. $B^0 = \mathcal{S}$. In this setting, the number of iterations to convergence matches the number of iterations to convergence in the standard value iteration setting. However, the the number of updated states per iteration decreases sharply, leading to a drastic decrease in the number of updates that we perform, especially for very sparsely connected graphs (see Figure 2 third row).
In the second variant, we choose a random subset of K states to update at the first iteration, then at each step, we choose a subset of K states to update based on the influence tree of the

previously updated states. If there are no states left to be updated, we again sample K states at random from the set of all states and continue until convergence. We note that this setting generally takes more iteration to converge than the standard VI setting, however the number of updates is much lower than in the standard VI method (see Figure 5). It is perhaps better to compare it to the Random-K VI method. In comparison to this case, we see not only a lower rate of updates per iteration (we may perform less than K updates per iteration in the tree based method), but we also see a *much* faster rate of convergence, particularly in sparsely connected MDPs. In one of the cases included, where $|\mathcal{A}| = 2$ from each state, the Random-K method takes over 2800 iterations (see Figure 1 for $k = 50, |\mathcal{A}| = 2$) whereas the tree based method takes around 500 (see Figure 5 for $k = 50, |\mathcal{A}| = 2$). In fact, once it "locks on" to an important state and traces it back, we see very very fast convergence. This method outperforms Random-K iteration in both updates per iteration as well as number of iterations and outperforms normal VI in the total number of updates.

We also compare Influence Tree VI while varying the state sampling size & action state space size vs state sampling size & action sampling size (see Figure 5 and Figure 4 respectively). We observe that when state sampling size increases then it takes less epochs for the method to converge, and the sparser the transitions are longer it takes to converge ((see Figure 5 columns). However, when on top of states sampling we also sample actions with replacement then the aforementioned trends are preserved but the process takes even longer to converge. Note that in order to keep the monotonicity of improvement of the estimates of a value function apart from sampling n_a actions we kept the last action that maximized our reward in the sample for the next iteration of that state.

Question 5

In general, we see that cyclic value iteration has slightly faster convergence than pure value iteration. We see this trend across sparsities of connections between states. If our MDP were a dag, and we could create a linearization of it, then running Cyclic VI would give us huge gains over regular VI. We can see the connection with the influence tree based methods as if we perform one iteration of cyclic VI in a linearized order, then we get the benefits of using the newest estimates of the value function without waiting for the new epoch to access more accurate value. Similarly in the influence tree, every iteration we bootstrap influenced state values based on the most recent estimates of the cost-to-go of the dependent states.

Question 6

We see a sharp improvement in the convergence of RPCyclic value iteration over regular value iteration and cyclic value iteration (see Figure 2). This improvement is especially pronounced when connections between states are sparse. This improvement is due to the

fact that randomly permuting the order of updates is more likely to approximate a good update order (like an influence tree based iteration) than picking a single "random" order and using that for the update sequence at each iteration like in regular Cyclic VI.

Comparison of all methods

All aforementioned observation of experiments are coherent among mazes of different sizes: $\{50 \times 50, 100 \times 100, 200 \times 200\}$ (see our GitHub repository for more plots: maze 100×100 https://github.com/sven-lerner/cme_307_project/blob/master/vi_100.ipynb, maze 200×200 https://github.com/sven-lerner/cme_307_project/blob/master/vi_200.ipynb).

Namely, when the sampled state size grows or the sparsity of the transition in the 2D maze decreases it leads to faster reward propagation. And as a result we observe less and less epochs before the convergence.

Let's take a look at the specific setting and compare the performance of all methods. Set $|\mathcal{A}| = 4$

- Standard VI: updates 2500 states at each iteration over 50 epochs, in total 125000 unit updates (see Figure 2)
- Random VI, $k = 250$: updates 250 states at each iteration over 350 epochs, in total 87500 unit updates (see Figure 1)
- Cyclic VI: updates 2500 states at each iteration over 40 epochs, in total 100000 unit updates (see Figure 2)
- Random Cyclic VI: updates 2500 states at each iteration over 24 epochs, in total 60000 unit updates (see Figure 2)
- Influence Tree VI (all states): updates on average 250 states at each iteration over 48 epochs, in total 12000 unit updates (see Figure 2)
- Influence Tree VI, $k = 250$: updates on average 250 states at each iteration over 50 epochs, in total 12500 unit updates (see Figure 5)
- Influence Tree VI, $k = 250$, sample 4 actions out of 8: updates on average 250 states at each iteration over 350 epochs, in total 87500 unit updates (see Figure 4)

Hence, when Influence Tree computed for all states (we keep all influenced states for an update rather than sample k states from them like in Influence Tree VI, $k = 4$) it is the most efficient way in terms of number of updates necessary for the convergence. However, if $|\mathcal{S}|$ is huge one epoch of Random Cyclic VI might be not feasible to compute, therefore we should use one of the state/action downsampling methods like Influence Tree VI or Random VI (see Figure 3), where Influence Tree VI has much faster convergence since it uses not only random states, but also such that propagate a signal from the previous iterations.

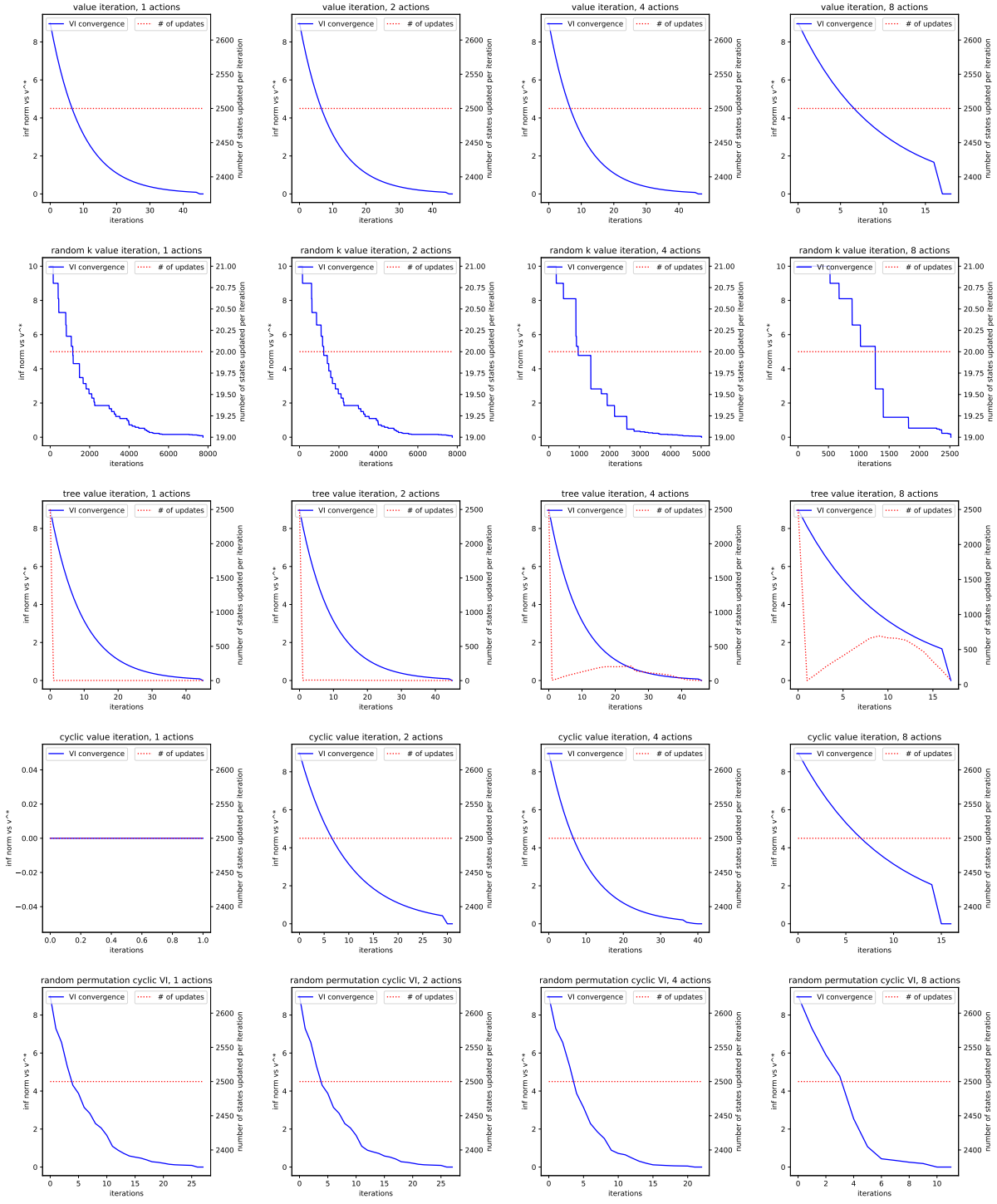

Figure 2: VI, Random VI, Influence Tree VI, Cyclic VI, Random Cyclic VI performance depending on the sparsity of transition probabilities $|\mathcal{A}|$. Maze size 50×50.

Appendix

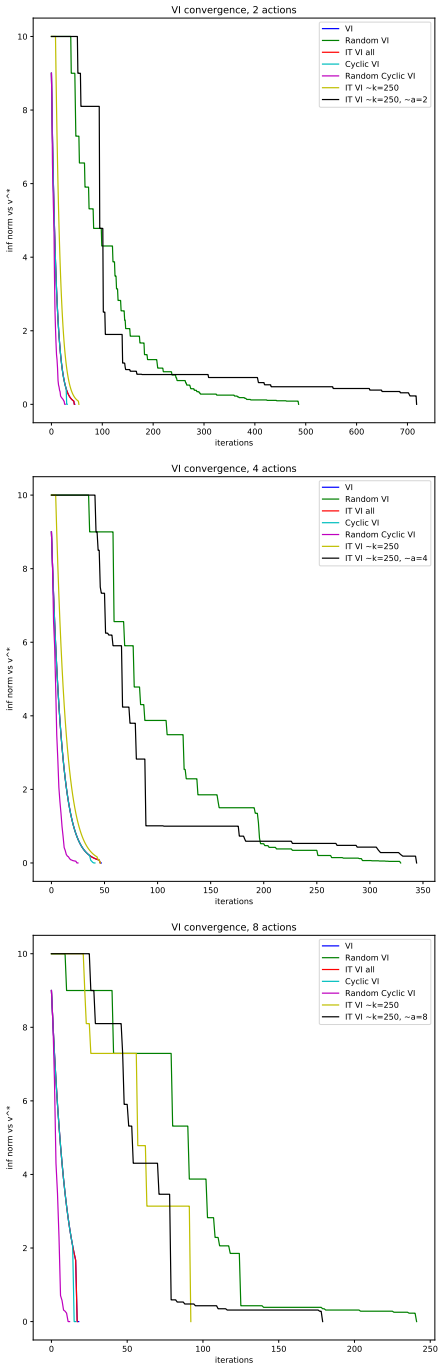

Figure 3: Comparison of VI convergence and value function quality for all methods. Maze size 50×50.

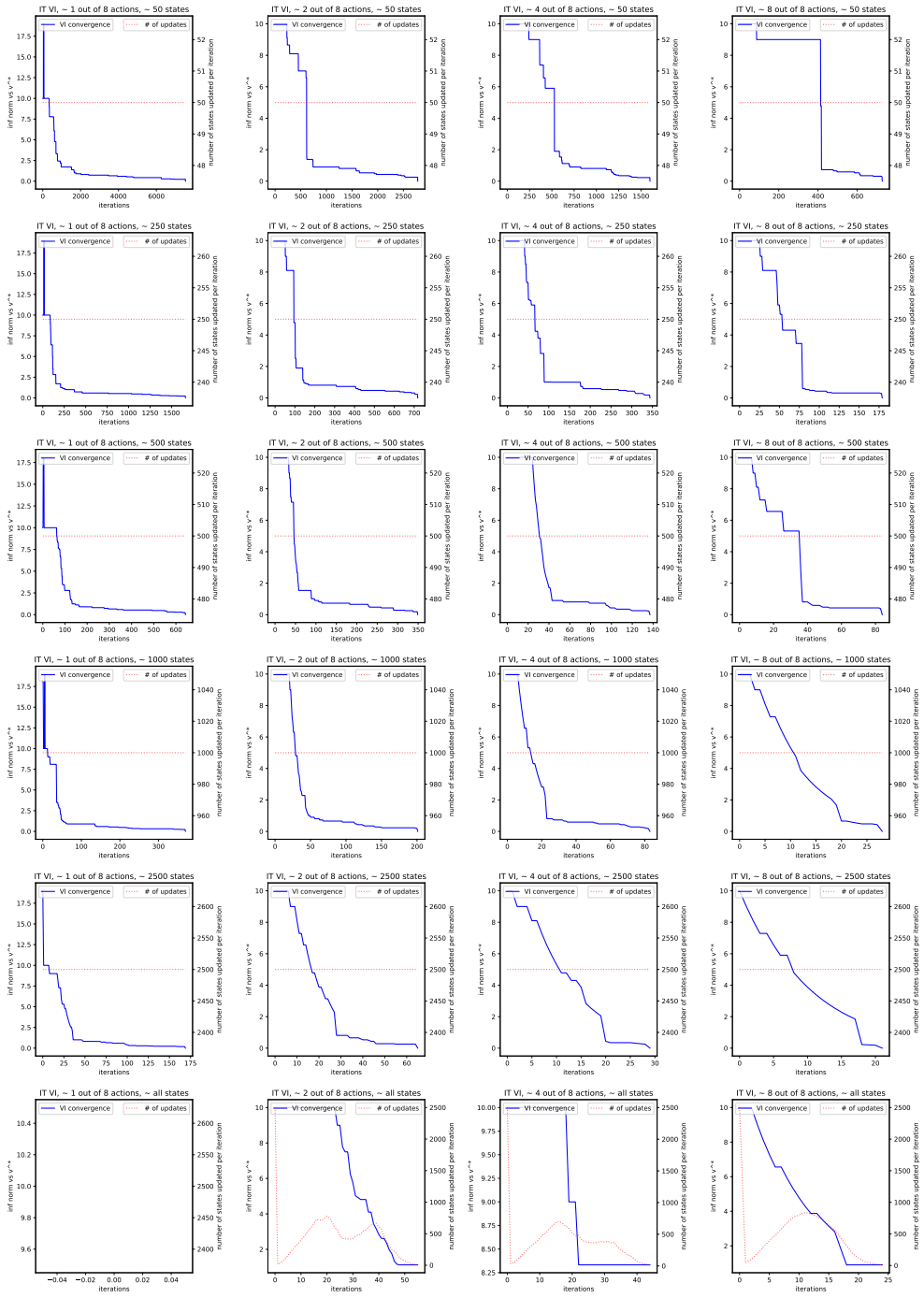

Figure 4: Influence Tree VI, dependency on the size of sampled states space k and sampled action space (out of total of $|\mathcal{A}| = 8$). Maze size 50×50.

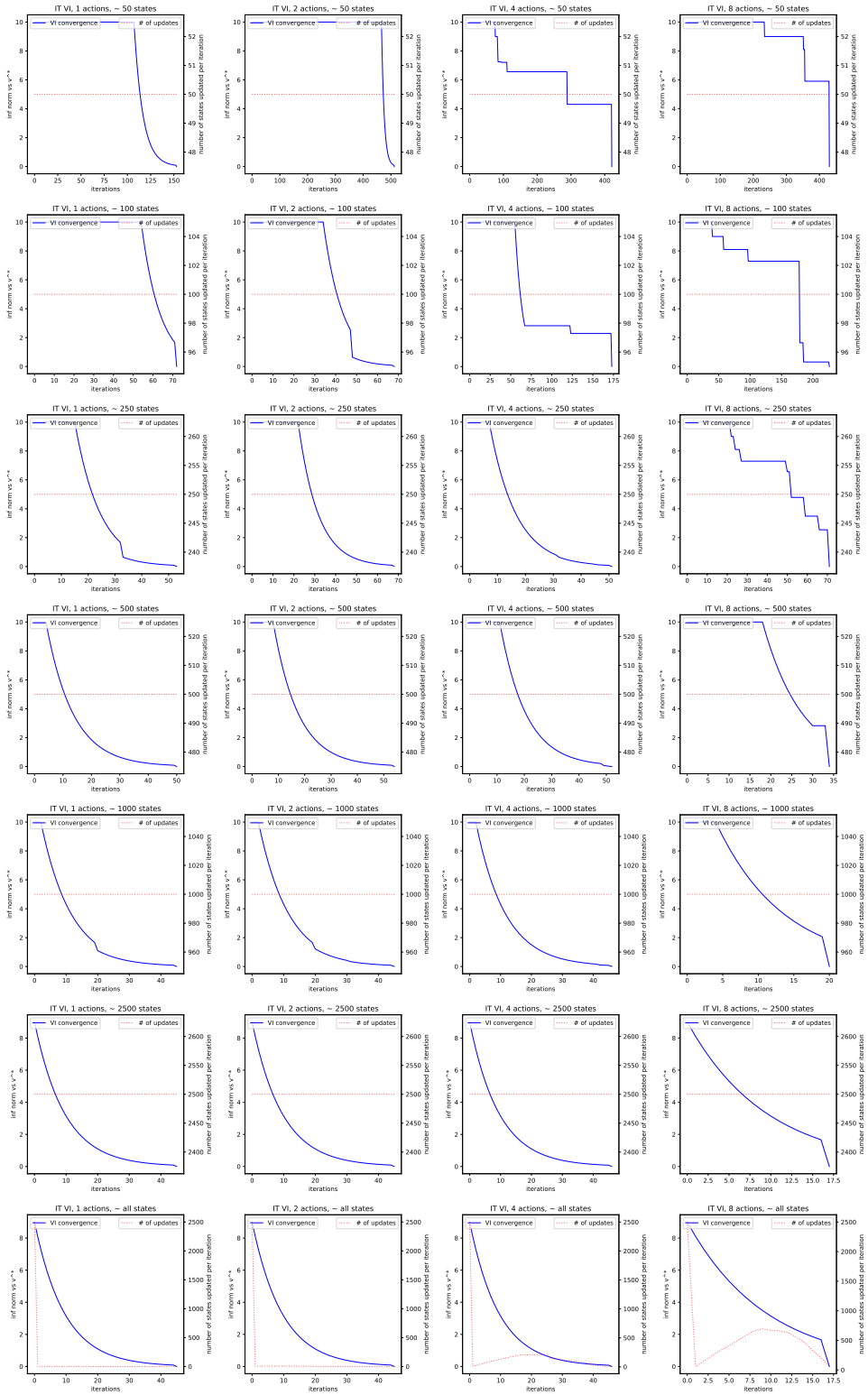

Figure 5: Influence Tree VI, dependency on the size of sampled states space k and action state space size $|\mathcal{A}|$ (actions are fixed along one experiment). Maze size 50×50.