# R08/Sarthak Mittal/200050129

February 15, 2023

**Markov Decision Processes (MDPs)** are an essential part of artificial intelligence applications that involve decision-making. **Policy Iteration (PI)** is a class of planning algorithms for MDPs, and the variants of it differ in the way they perform **"switching"**. This paper gives further insight into obtaining **strong upper bounds** on the number of iterations and achieves a significant improvement over previous results related to lower and upper bounds for Random PI (RPI), Howard's PI (HPI), and shows the tightest yet upper bound for PIs using a randomized variant of **Batch-Switching PI (BSPI)**.

The MDP framework used for the analysis assumes that the state and action space are both **finite**, and policies are **stationary, deterministic, and Markovian**, and $P$ and $R$ are given as tables. The paper describes the PI method and how variations differ in terms of the "switching rule" they use. Their results also carry over to solving **Acyclic Unique Sink Orientations (AUSO)** problems, and the AUSOs resulting from 2-action MDPs also satisfy the **Holt-Klee conditions** due to being linked to **linear programs**.

In the previous analysis, the tightest upper bound was obtained by bounding the number of policies eliminated in each iteration. The results suggested an upper bound of $\mathcal{O}(k^n/n)$ for HPI, and $\mathcal{O}(((1+2/logk)(k/2))^n)$ for RPI. The recent analysis of BSPI with a batch size of 7 showed an upper bound of $\mathcal{O}(k^{0.7207n})$. The analysis of the **RSPI** algorithm, which picks an improving action **uniformly at random**, results in an upper bound of $\mathcal{O}((2+ln(k-1))^n)$.

**Upper Bounds:** The first result they show is that the sequence of the sizes of modification sets for each state is unique for a given policy, thus establishing a **bijection** from the set of **policies** to the set of **"improvement sequences"**. To improve the upper bound of **RPI**, they choose the improvable action uniformly at random, which requires polynomial-time operations. The derivation of the upper bound follows from first bounding the number of **small-improvement policies**, then lower bounding the number of policies **skipped**, resulting in an upper bound of $\mathcal{O}(k^{n/2}H_k^{(n-1)/2})$. For the upper bound of **HPI**, they use a variant in which the improving action is picked uniformly at random. Following a similar analysis, the upper bound obtained is $\mathcal{O}((2k)^{n/2}H_k^{(n-1)/2})$.

The original **BSPI** algorithm was analyzed using **HPI** within the batches, **enumerating** all possible AUSOs of dimensions up to 4. In this paper, they use **RPI** within the batches, and RPI dominates HPI in the expected number

of iterations. The upper bound obtained is $\mathcal{O}(1.6001^n)$ for 2-action MDPs.

To obtain the lower bound for RPI, they use the presence of **dummy states** to eliminate policies, represent policies as **bit strings**, and the expected number of policies evaluated by RPI turns out to be at least $(n+1)/2$.

In conclusion, they show their results using experiments. In practice, **HPI** seems to work better than RPI. They claim the disparity is due to **loose bounds** and their **choice** of MDPs. The analysis also does not explicitly use the **properties** of MDPs.