A Tighter Analysis of Randomised Policy Iteration

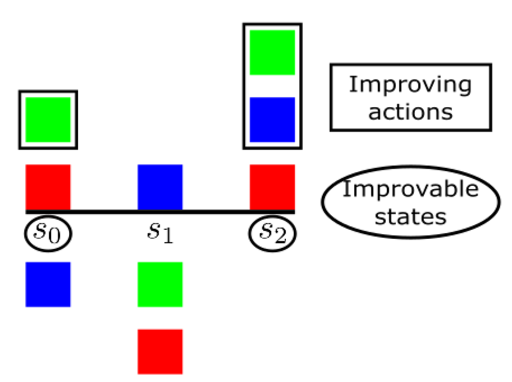
Meet Taraviya and Shivaram Kalyanakrishnan | {mtaraviya, shivaram}@cse.iitb.ac.in

Department of Computer Science and Engineering, IIT Bombay

Introduction

- **Markov Decision Processes**
  - model "agent-environment-reward" systems.
  - consist of States, Actions, Transition Probabilities, Rewards and Discount factor (γ).
- A **policy** decides what action the agent takes at each state.
- Goal: To find the policy that maximizes "long-term" reward (expected infinite discounted reward).
- *State-value* $V^\pi(s)$: long-term reward starting from s, following policy π.
- *Action-value* $Q^\pi(s, a)$: long-term reward starting from s, taking a and following policy π thereafter.
- V^π, Q^π: *evaluated* by solving a system of linear equations.
- If $Q^\pi(s, a) > V^\pi(s)$: we say s is an *improvable state* and a is an *improving action* at s for policy π.

- **Policy Iteration**
  - Start with an initial policy
  - While the current policy is not optimal:
    (1) Evaluate the policy;
    (2) Select one or more improvable states;
    (3) Select an improving action at each of these states;
    (4) Update the policy



- Different selection strategies → different PI variants.

Some PI variants

- **Howard's PI**
  - Earliest PI variant, introduced by [Howard, 1960].
  - *Every* improvable state is improved; the improving actions are selected *arbitrarily*.
- **Randomised PI**
  - Introduced by [Mansour and Singh, 1999].
  - The set of states to be improved is selected *randomly* from all non-empty subsets of the set of improvable states; the improving actions are selected *arbitrarily*.
- **Batch-switching PI**
  - Introduced by [Kalyanakrishnan et al., 2016].
  - Provides a scheme to translate upper bounds for constant -sized MDPs to general MDPs.
  - States divided into batches of size b; states only within a single batch are allowed to be improved.
  - Within a batch, selection of states to be improved and improving actions can be dictated by some other algorithm (like HPI or RPI).

Contributions

Variant	Previous	This paper
HPI	$O(\frac{k^n}{n})$	$(O(k \log k))^{n/2}$ for HPI-R
RPI	$O(((1+\frac{2}{\log_2 k})\frac{k}{2})^n)$	$(O(k \log k))^{n/2}$ for RPI-UIP
	–	$\Omega(n)$ for $k = 2$
BSPI	$O(k^{0.7207n})$	$O(k^{0.7019n})$ for BSPI(HPI)
		$O(k^{0.6782n})$ for BSPI(RPI)

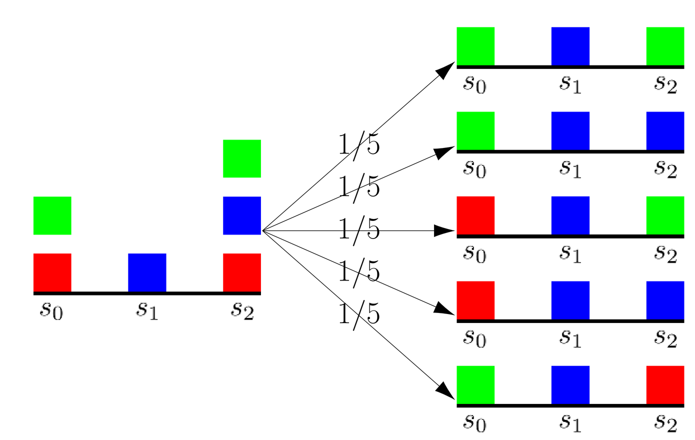
A lemma on the structure of policy space

- *Improvement sequence*: the sequence consisting of the number of improving actions for a policy at each state.

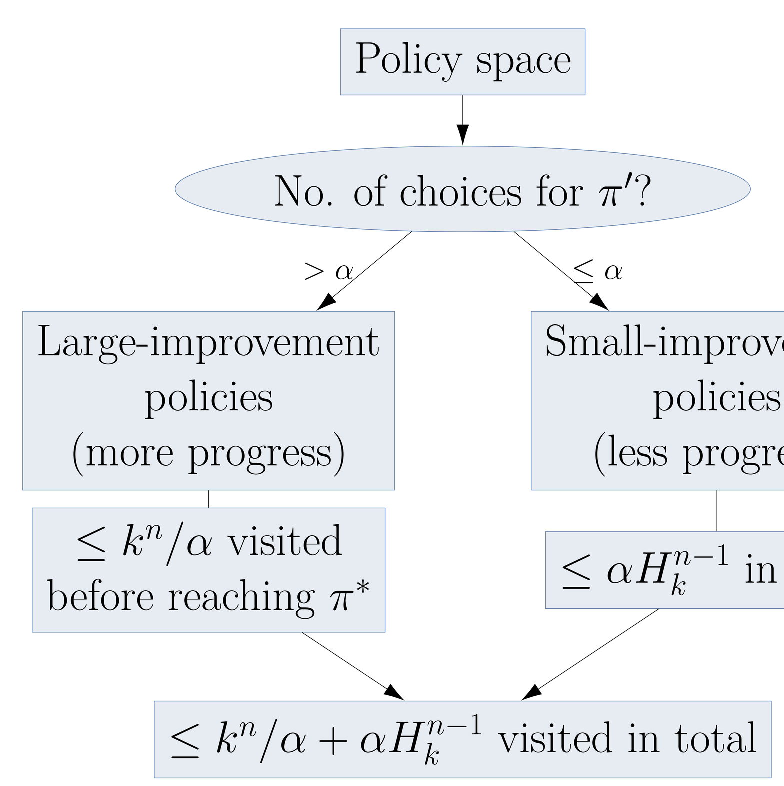
Main Result: *The map from policies to their improvement sequences is a bijection.*

- Was discovered for $k = 2$ by [Gupta and Kalyanakrishnan, 2017]; we generalized to $k \geq 2$.

RPI-UIP upper bound



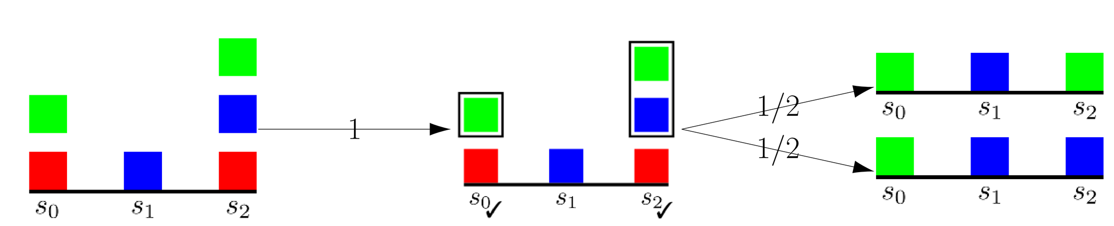
RPI-UIP



Proof structure

RPI-UIP takes at most $O(k^{n/2} H_k^{(n-1)/2})$ iterations.

HPI-R upper bound



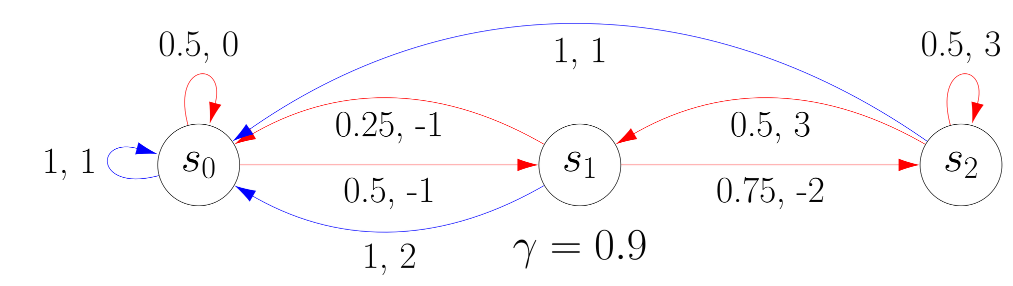
HPI-R

Similar proof structure, but $\Omega(\alpha/2^n)$ policies are skipped at large-improvement policies instead of $\Omega(\alpha)$.

HPI-R takes at most $O(2^{n/2} k^{n/2} H_k^{(n-1)/2})$ iterations.

2-action MDPs and AUSOs

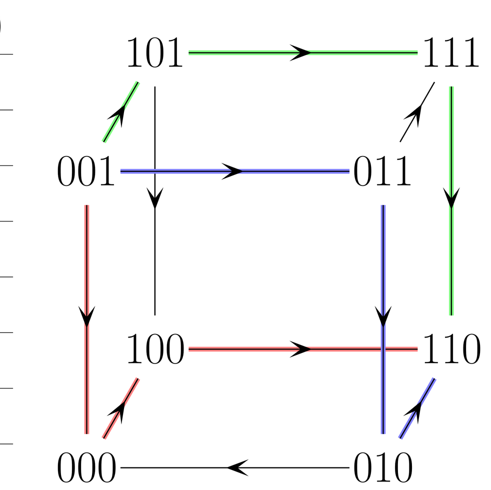
- Policies of a n-state 2-action MDP can be arranged as the vertices of n-dimensional Acyclic Unique Sink Orientation.
- n-AUSOs produced in this way are also known to satisfy the Holt-Klee property: there are n inner-vertex-disjoint paths from the source to the sink [Holt and Klee, 1999].
- By running RPI and HPI on all Holt-Klee 4-AUSOs, we found that they take at most 6.5544 and 7 iterations resp. on 4-state 2-action MDPs.
- These translate to upper bounds of 1.6001^n and 1.6266^n for BSPI (RPI) and BSPI (HPI) resp. on n-state 2-action MDPs
- Using ideas from [Gupta and Kalyanakrishnan, 2017], we get deterministic and randomised PI algorithms taking $O(k^{0.7019n})$ and $O(k^{0.6782n})$ iterations resp.



(a)

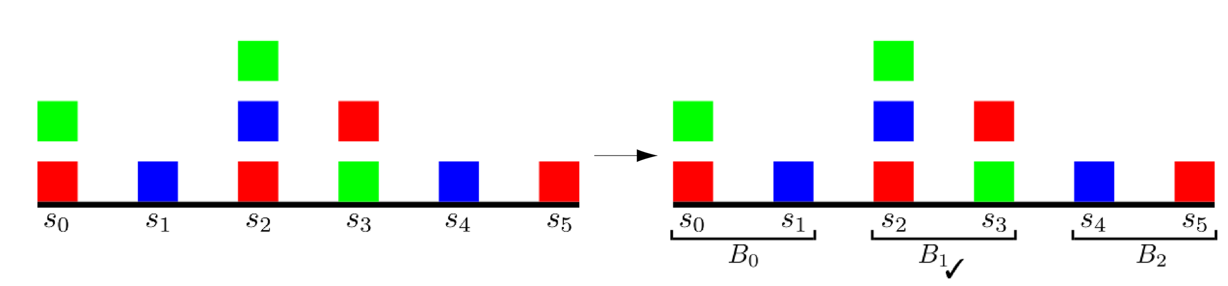
π	$T^\pi(s_0)$	$T^\pi(s_1)$	$T^\pi(s_2)$
000	{1}	ϕ	ϕ
001	{1}	{1}	{0}
010	{1}	{0}	ϕ
011	{1}	ϕ	{0}
100	ϕ	{1}	ϕ
101	ϕ	{1}	{0}
110	ϕ	ϕ	ϕ
111	ϕ	ϕ	{0}

(b)

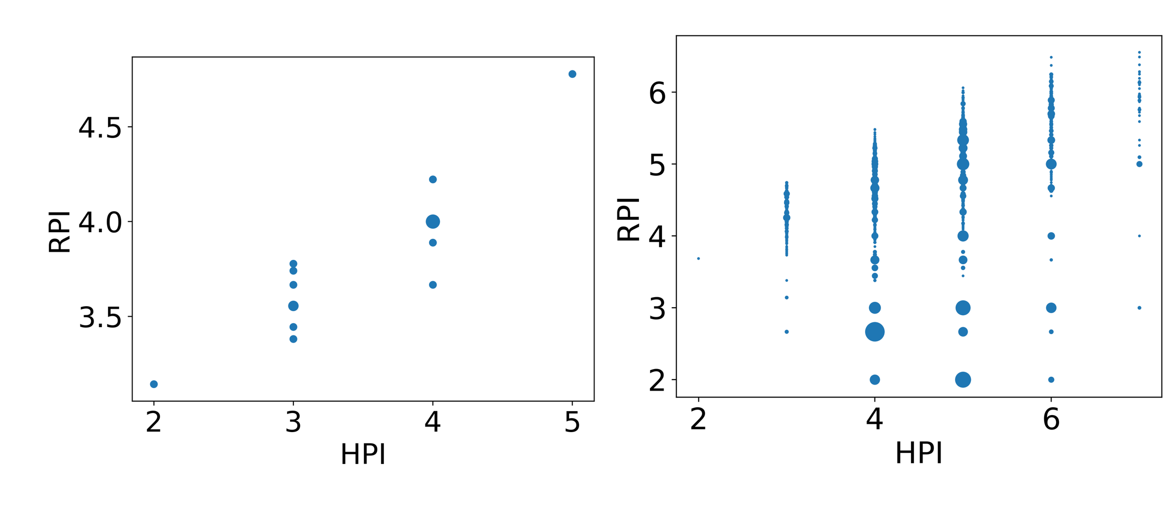


(c)

(a) A 3-state 2-action MDP; $A = \{0, 1\}$ (b) Improving actions for each state, for each policy, demonstrating the bijection lemma (c) The 3-AUSO corresponding to the above MDP

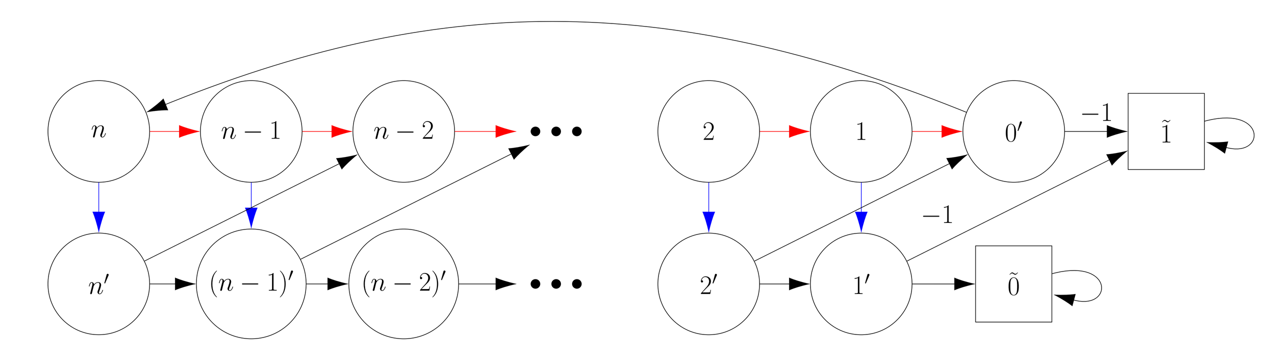


BSPI template



No. of iterations taken by HPI and RPI on AUSOs; Larger circles corresponding to multiple AUSOs

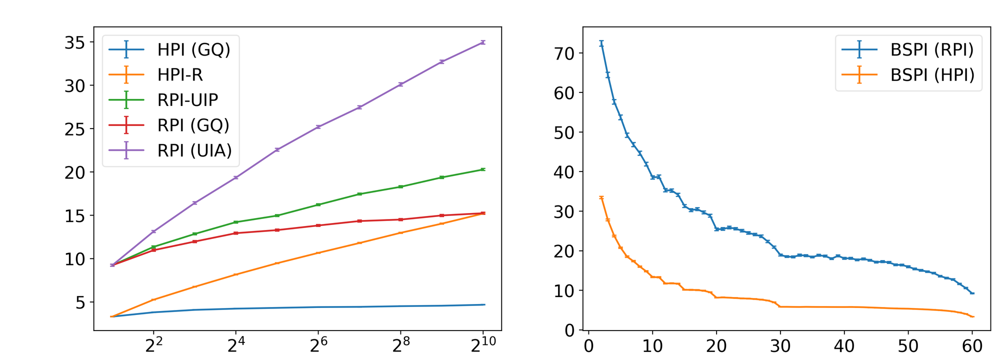
RPI lower bound for $k = 2$



MDP M_n used to prove the RPI lower bound; $A = \{0, 1\}$; $\gamma = 1$

Starting from $\pi^0 = 0^n$, RPI takes at least $\frac{n+1}{2}$ iterations on M_n in expectation.

Experiments



(a) (b)

- Each graph plots an *average* over 500 randomly generated 60-state MDPs.
- Figure (a) compares performance of HPI and RPI variants as a function of k. Greedy action-selection rule is found to work better in practice than a randomised one.
- Figure (b) plots the effect of batch size b on the number taken by BSPI (HPI) and BSPI (RPI). For both variants, the no. of iterations drops fairly consistently with increase in b.
- Howard's improvable state selection rule performs better than randomised selection, even within the framework of BSPI.

References

[Gupta and Kalyanakrishnan, 2017] Gupta, A. and Kalyanakrishnan, S. (2017). Improved strong worst-case upper bounds for mdp planning. In *IJCAI-17*.

[Holt and Klee, 1999] Holt, F. and Klee, V. (1999). A proof of the strict monotone 4-step conjecture. *Contemporary Mathematics.*

[Howard, 1960] Howard, R. A. (1960). *Dynamic programming and Markov processes.*

[Kalyanakrishnan et al., 2016] Kalyanakrishnan, S., Mall, U., and Goyal, R. (2016). Batch-switching policy iteration. In *IJCAI-16.*

[Mansour and Singh, 1999] Mansour, Y. and Singh, S. (1999). On the complexity of policy iteration. In *UAI-99.*