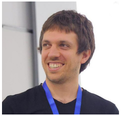
Value Iteration with Guessing for Markov Chains and Markov Decisions Processes

Krishnendu Chatterjee

Mahdi Jafariraviz

Raimundo Saona



Jakub Svoboda

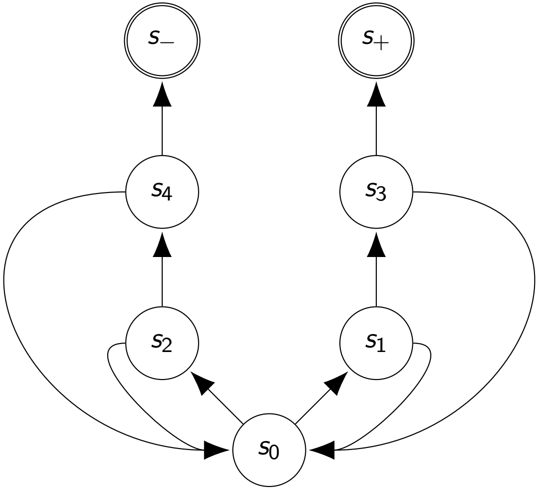
TACAS 2025, Hamilton, Canada

ISTA Institute of Science and Technology Austria

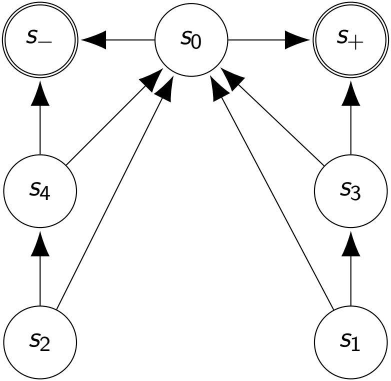
Incorporating Binary Search into Value Iteration in MDPs:

Practical implementation

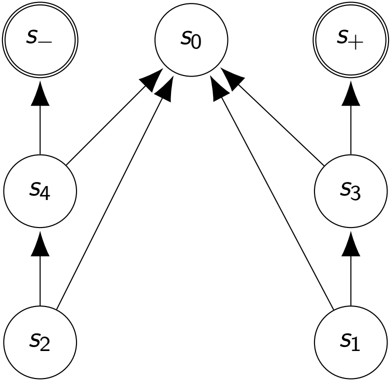
Stochastic system



Stochastic system, after guessing



Stochastic system, after guessing



Verifying a guess

Lemma

Consider a game G, a state $s \in S$ and a guess $\gamma \in [0,1]$.

$$\mathrm{Update}(s, (\gamma, \mathsf{val}_{G[s=\gamma]})) > \gamma \quad \Leftrightarrow \quad \mathsf{val}_G(s) > \gamma \,.$$

Approximately verifying a guess

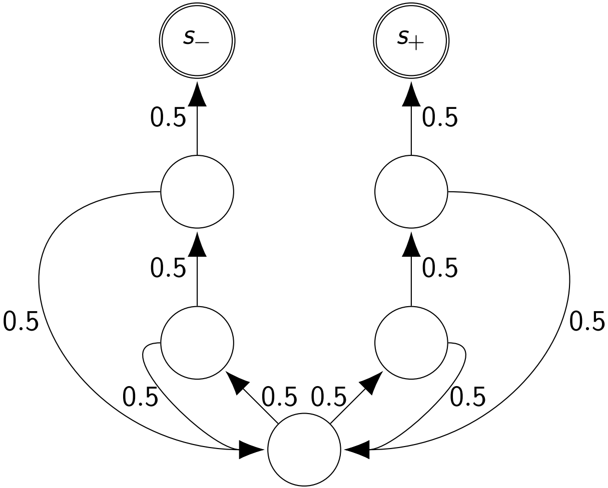
Lemma

Consider a game G, a state $s \in S$, a guess $\gamma \in [0,1]$, and $\varepsilon > 0$.

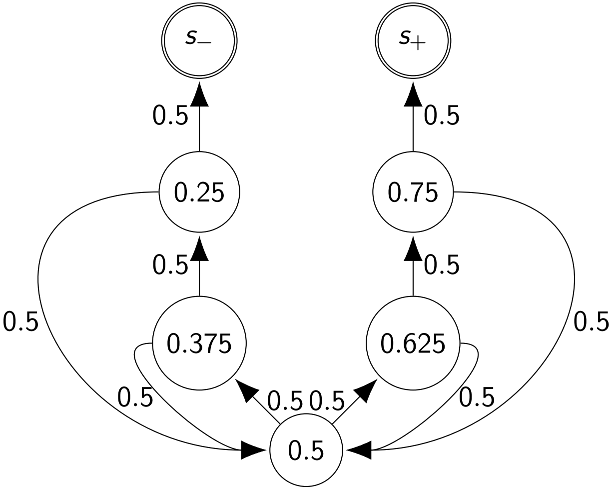
$$\mathrm{Update}(s, (\gamma, \mathsf{val}_{G[s=\gamma]})) + p_{\min}^{|S|}\,\varepsilon > \gamma \quad \Rightarrow \quad \mathsf{val}_G(s) > \gamma - \varepsilon\,.$$

Can we speed up Value Iteration?

Markov Chain



Reachability values



Update operator

For $v \in \mathbb{R}^S$,

$$\mathrm{Update}(v)(s) := \max_{a \in A} \left\{ \sum_{s' \in S} p(s' \mid s, a)\, v(s') \right\}$$

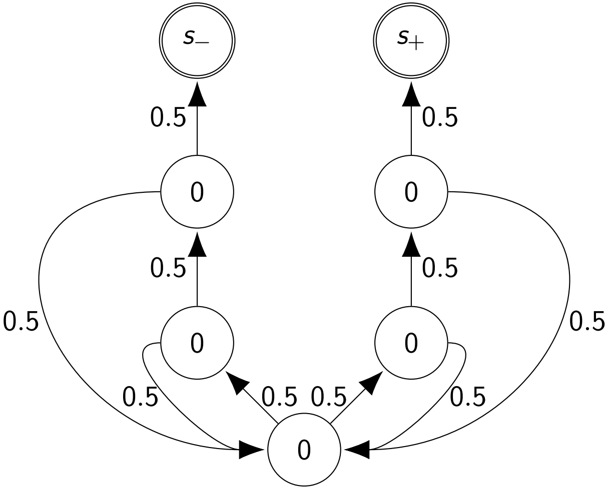
The reachability value is the least solution with

$$\mathsf{val}(s_+) = 1 \qquad\qquad \mathsf{val} = \mathrm{Update}(\mathsf{val})$$

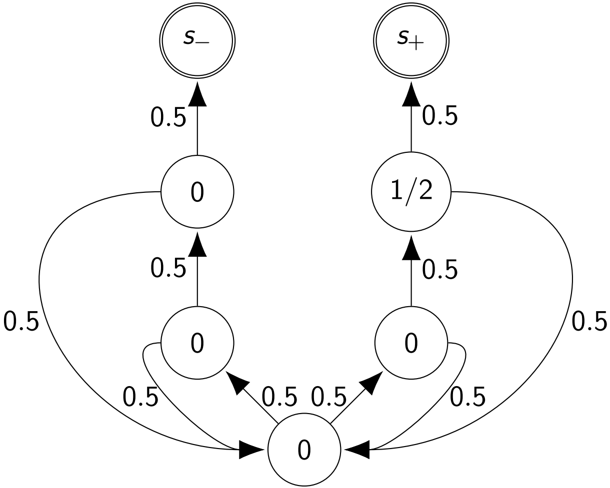
Moreover, if $v_0(s) = \mathbb{1}[s = s_+]$ and $v_{i+1} := \mathrm{Update}(v_i)$, then

$$(v_i) \xrightarrow[i \to \infty]{} \mathsf{val}$$

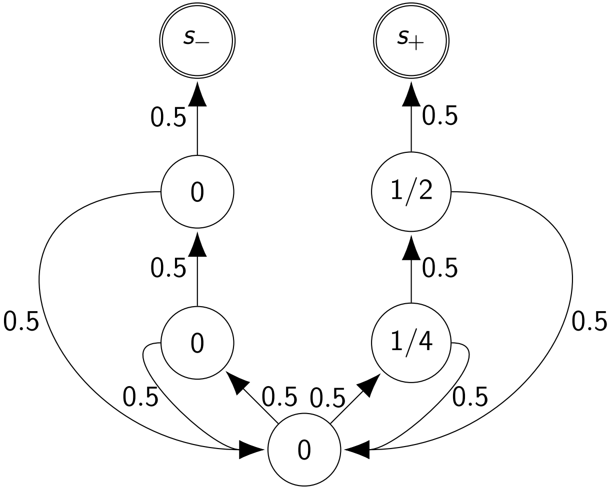
Value Iteration



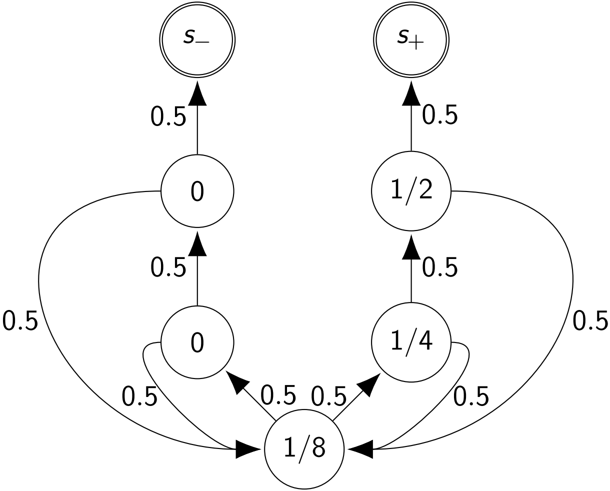
Value Iteration



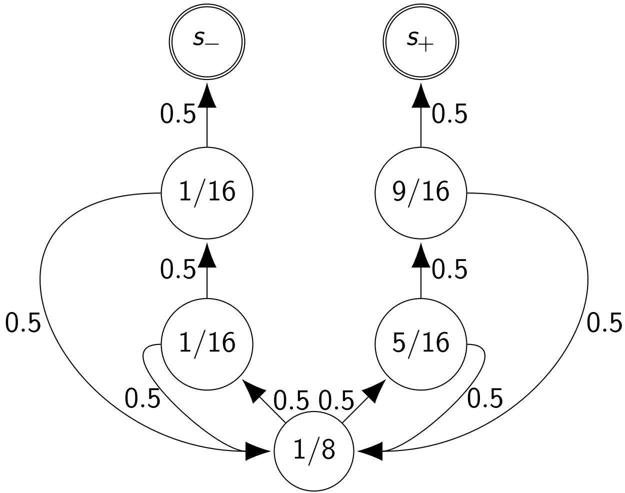
Value Iteration



Value Iteration



Value Iteration



Value Iteration speed

Lemma

The speed of convergence is driven by

$$\|v_{|S|i} - \mathrm{val}\|_\infty \leq \left(1 - p_{\min}^{|S|}\right)^i \|v_0 - \mathrm{val}\|_\infty$$

So, to have an ε*-approximation of* val, *Value Iteration requires many iterations:*

$$(-\log(\varepsilon)|S|/p_{\min})^{\Omega(|S|)}$$

In other words, **exponentially** *many.*

Value Iteration speed: Levels

Lemma

The speed of convergence is driven by

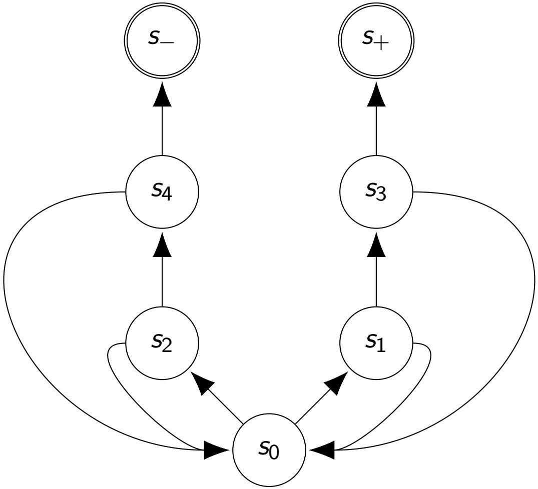
$$\|v_{Li} - \mathsf{val}\|_\infty \leq \left(1 - p_{\min}^L\right)^i \|v_0 - \mathsf{val}\|_\infty$$

where $L \leq |S|$ is the maximum graph distance to the target or sink. An ε-approximation of val *requires many iterations:*

$$(-\log(\varepsilon) L / p_{\min})^{\Omega(L)}$$

In other words, **exponentially** *many.*

Stochastic system



Value Iteration speed, improved

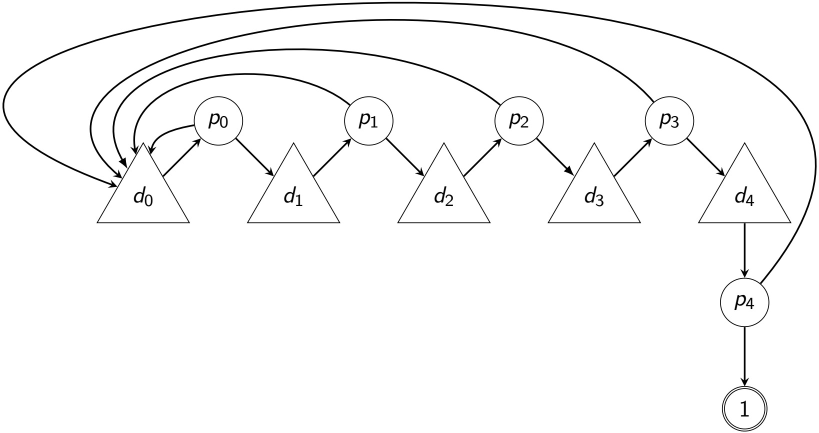
Theorem

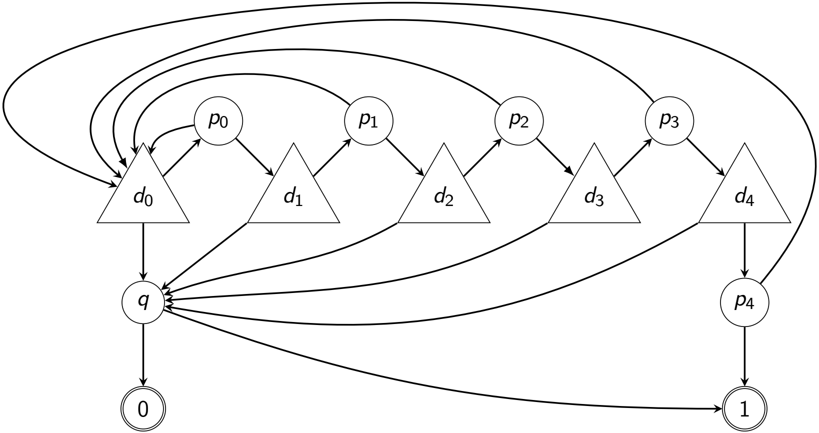
To have an ε-approximation of val*, we require fewer iterations:*

$$(-\log(\varepsilon)|S|/p_{\min})^{\mathcal{O}(\sqrt{|S|})}$$

In other words, **subexponentially** *many.*

MDPs


p_0
p_1
p_2
p_3
d_0
d_1
d_2
d_3
d_4
p_4
1


p_0
p_1
p_2
p_3
d_0
d_1
d_2
d_3
d_4
q
p_4
0
1

Value Iteration is not accelerated
by having low distance to the targets in MDPs.

What about practical speedups?

Practical implementation

Guessing, solving, and verifying implies a recursive algorithm.
Recursion depth is given by the number of states guessed.
Guessing more than 3 states is impractical.

Practical considerations:

- Early Verification of a Guess
- Reusing Bounds
- Picking the Guessed States
- Benefiting from VI

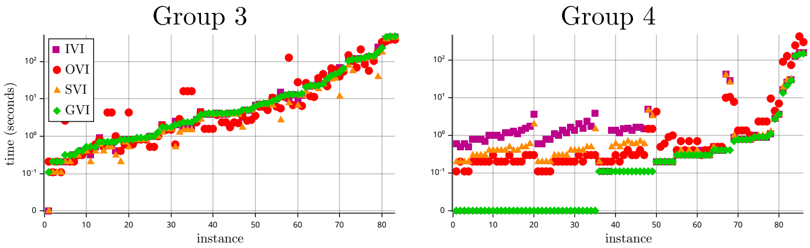
We consider the Quantitative Verification Benchmark Set (QVBS).
The following VI-based variants were considered.

- Interval VI
  Classic two-sided VI
- Optimistic VI
  Candidates for upper or lower bounds are proposed while Interval VI is running
- Sound VI
  Deduce upper bounds based on lower bounds
- Guessing VI
  Our algorithm

Grouping results

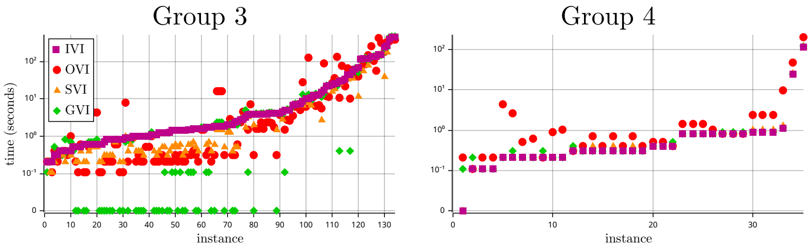
- Group 1
  All algorithms are fast, i.e., they take at most 0.1 seconds.
  170 instances.
- Group 2
  The fastest and slowest algorithms
  are only at most 1.10 times of each other.
  135 instances.
- Group 3
  Guessing VI is not the fastest approach.
  83 instances.
- Group 4
  All other instances not considered before.
  86 instances.

Performance: Guessing VI

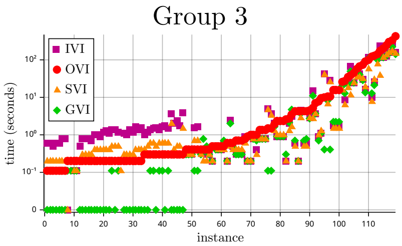


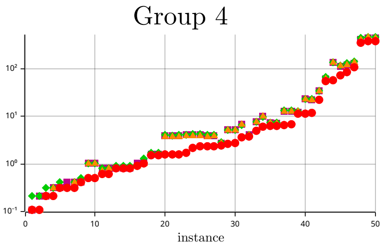
Time in seconds of all algorithms over instances in Groups 3 and 4 in increasing order **according to GVI** and displayed in logarithmic scale.

Performance: Interval VI



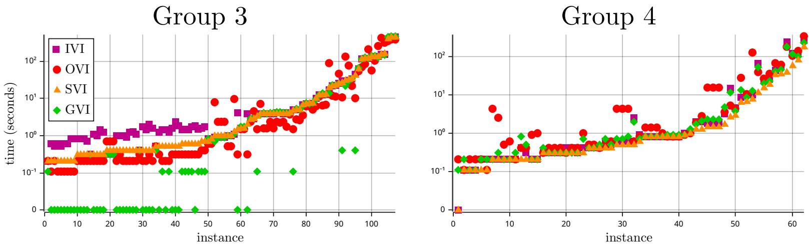
Time in seconds of all algorithms over instances in Groups 3 and 4 in increasing order **according to IVI** and displayed in logarithmic scale.





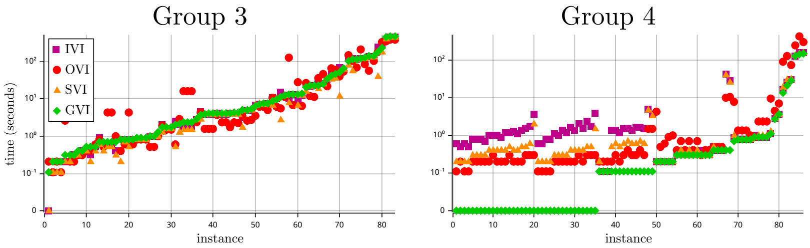
Time in seconds of all algorithms over instances in Groups 3 and 4 in increasing order **according to OVI** and displayed in logarithmic scale.

Performance: Sound VI



Time in seconds of all algorithms over instances in Groups 3 and 4 in increasing order **according to SVI** and displayed in logarithmic scale.

Performance: Guessing VI



Time in seconds of all algorithms over instances in Groups 3 and 4 in increasing order **according to GVI** and displayed in logarithmic scale.

Thank you!