Incremental Policy Iteration with Guaranteed Escape from Local Optima in POMDP Planning

Marek Grześ
School of Computing, University of Kent, UK

Pascal Poupart
David Cheriton School of Computer Science, University of Waterloo, Canada

University of Kent

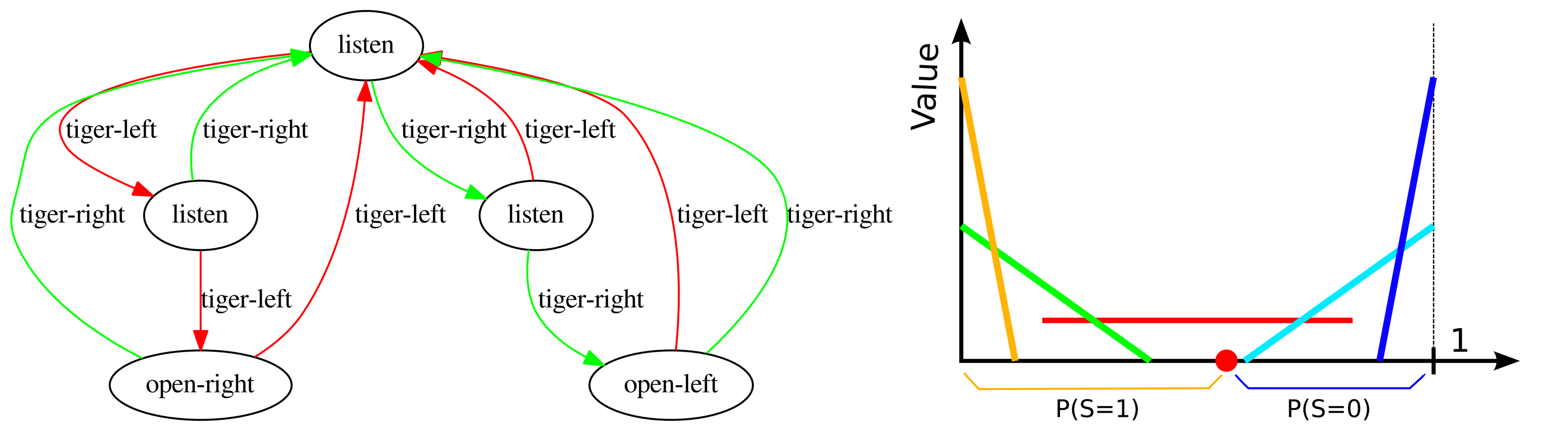
UNIVERSITY OF WATERLOO

AAMAS-15

Motivation and Contribution

- Finite-state controllers (FSCs) are the most energy efficient POMDP policies (see Grzes *et al.* "Energy Efficient Execution of POMDP Policies.") which shows their suitability for mobile applications
- Efficient and robust algorithms that compute small policies/controllers become desirable
- We investigate incremental methods that guarantee the escape from local optima
- We push the understanding and the performance of policy iteration for POMDPs to the point that for the first time they are competitive with the state-of-the-art point-based methods

Finite-state Controllers for POMDPs



Node Improvement in Bounded Policy Iteration (BPI)

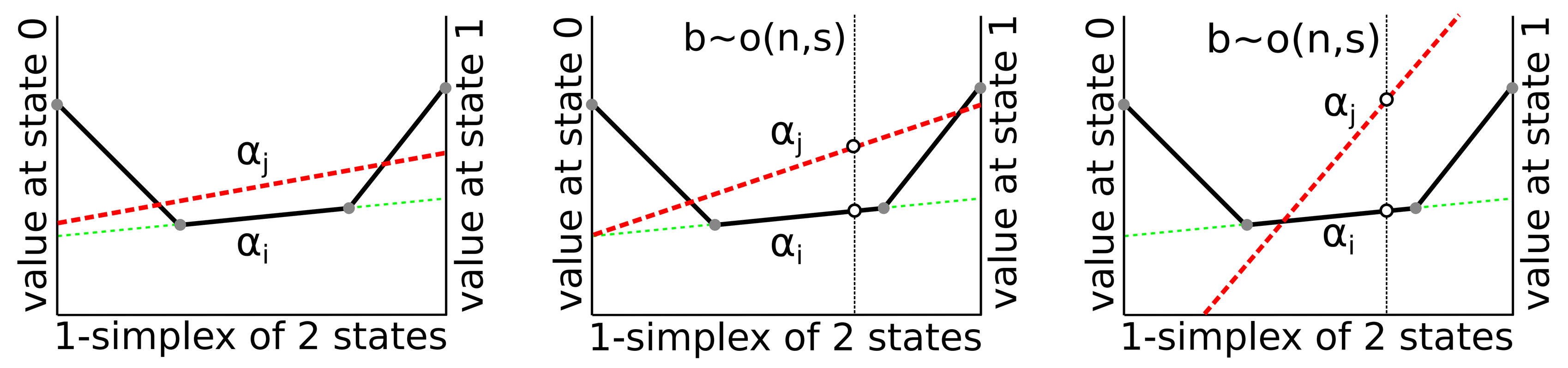


Figure 1 : Alpha vectors α_j for improved nodes.

The Need to Escape Local Optima

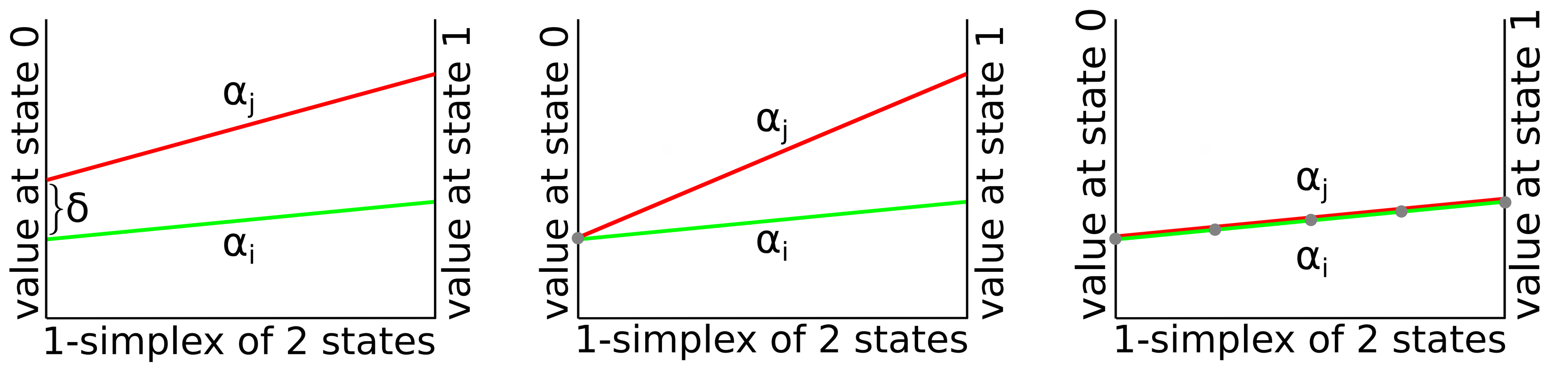


Figure 2 : Possible node improvements computed in BPI; α_i corresponds to an old node and α_j to a new node.

Find a New Node with Maximal Improvement Over Entire Belief Simplex

The naïve way to compute the exact DP update for POMDPs is to enumerate all possible alpha vectors. We can compute the set $\mathbf{\Gamma^{a,o}}$ of vectors $\mathbf{V^{a,o}_n(s)}$ for each $\langle \mathbf{a}, \mathbf{o} \rangle$ pair by applying a DP update:

$$\mathbf{\Gamma^{a,o}} \leftarrow \mathbf{V^{a,o}_n(s)} = \frac{\mathbf{R^a(s)}}{\mathbf{|O|}} + \gamma \sum_{\mathbf{s' \in S}} \mathbf{P(o|s',a)P(s'|a,s)V^{\pi}_n(s')}, \forall_{\mathbf{n}}$$

Having $\mathbf{V^{a,o}_n(s)}$ for each $\langle \mathbf{a}, \mathbf{o} \rangle$ pair, we can formulate the following optimisation problem:

$$\begin{aligned}
\text{max: } & \textstyle\sum_{a,n',o,s} \mathbf{w(s)P(n',a|o)V^{a,o}_{n'}(s)} - \beta \\
\text{s.t. } & \textstyle\sum_s \mathbf{w(s) = 1}; \sum_{n',a} \mathbf{P(n',a|o) = 1}; \\
& \forall_{a,o_1,o_2} \textstyle\sum_{n_1} \mathbf{P(n_1,a|o_1)} = \sum_{n_2} \mathbf{P(n_2,a|o_2)} \\
& \forall_n \beta \geq \textstyle\sum_s \mathbf{w(s)V^{\pi}_n(s)}; \\
& \forall_s \mathbf{w(s) \in R}; \forall_{n',a,o} \mathbf{P(n',a|o) \in [0,1]}
\end{aligned}$$

Figure 3 : A quadratic optimization program to search for a new node that provides maximal improvement at the entire belief simplex; the belief $\mathbf{w}$ (witness belief) is the belief at which the improvement happens. Decision variables are the witness belief $\mathbf{w}$, the current value β at belief $\mathbf{w}$, and node parameters $\mathbf{P(n',a|o)}$ which, when interpreted as probabilities, correspond to $\mathbf{P(n'|o)P(a)}$.

Optimal Solution to the Escape Problem

Theorem

There always exists an optimal solution to the quadratic problem shown in Fig. 3 that is integral $\forall_{\mathbf{n',a,o}} \mathbf{P(n',a|o)}$, *i.e., there exists an optimal solution that corresponds to a deterministic node.*

Our Algorithm

$$\begin{aligned}
\text{max: } & \textstyle\sum_{a,n',o,s} \mathbf{y(s,n',a,o)V^{a,o}_{n'}(s)} - \beta \\
\text{s.t. } & \textstyle\sum_s \mathbf{w(s) = 1}; \sum_{n',a} \mathbf{P(n',a|o) = 1}; \\
& \forall_{a,o_1,o_2} \textstyle\sum_{n_1} \mathbf{P(n_1,a|o_1)} = \sum_{n_2} \mathbf{P(n_2,a|o_2)} \\
& \forall_n \beta \geq \textstyle\sum_s \mathbf{w(s)V^{\pi}_n(s)}; \\
& \forall_s \mathbf{w(s) \in R}; \forall_{n',a,o} \mathbf{P(n',a|o) \in \{0,1\}}; \\
& \forall_{s,a,o,n'} \mathbf{0 \leq y(s,a,o,n') \leq P(n',a|o)}; \\
& \forall_{s,a,o,n'} \mathbf{w(s) + P(n',a|o) - 1 \leq y(s,a,o,n') \leq w(s)};
\end{aligned}$$

Real variables Integer variables

Figure 4 : The McCormick transformation of the quadratic program in Fig. 3.

- Thanks to the above theorem, McCormick relaxation finds an optimal, deterministic node
- MILP is intractable, but we don't need the optimal solution
- Even a linear relaxation of our MILP can be sufficient (see the paper for interesting properties)

Practial Implementation with Fast Heuristics

Algorithm 1: IPI(-LP): Incremental Policy Iteration for POMDPs.

```
Data: POMDP
Result: FSC for POMDP
FSC.N ← {n_1} ;                                  /* the first node */
FSC.N ← FSC.N ∪ {n_2} ;                          /* the second node */
while impr = true do
    Policy evaluation
    for n ∈ FSC.N do
        impr ←ImproveNode(FSC, n);               /* DP or LP */
    if ¬impr then                                /* escape is required */
        impr ←OnPolicyLH(FSC)
        if ¬impr then
            impr ←BestOf(OffPolicyLH, Split, Corner)
            if ¬impr then
                impr ←MILPEscape(FSC)
Prune dominated nodes
```

Results

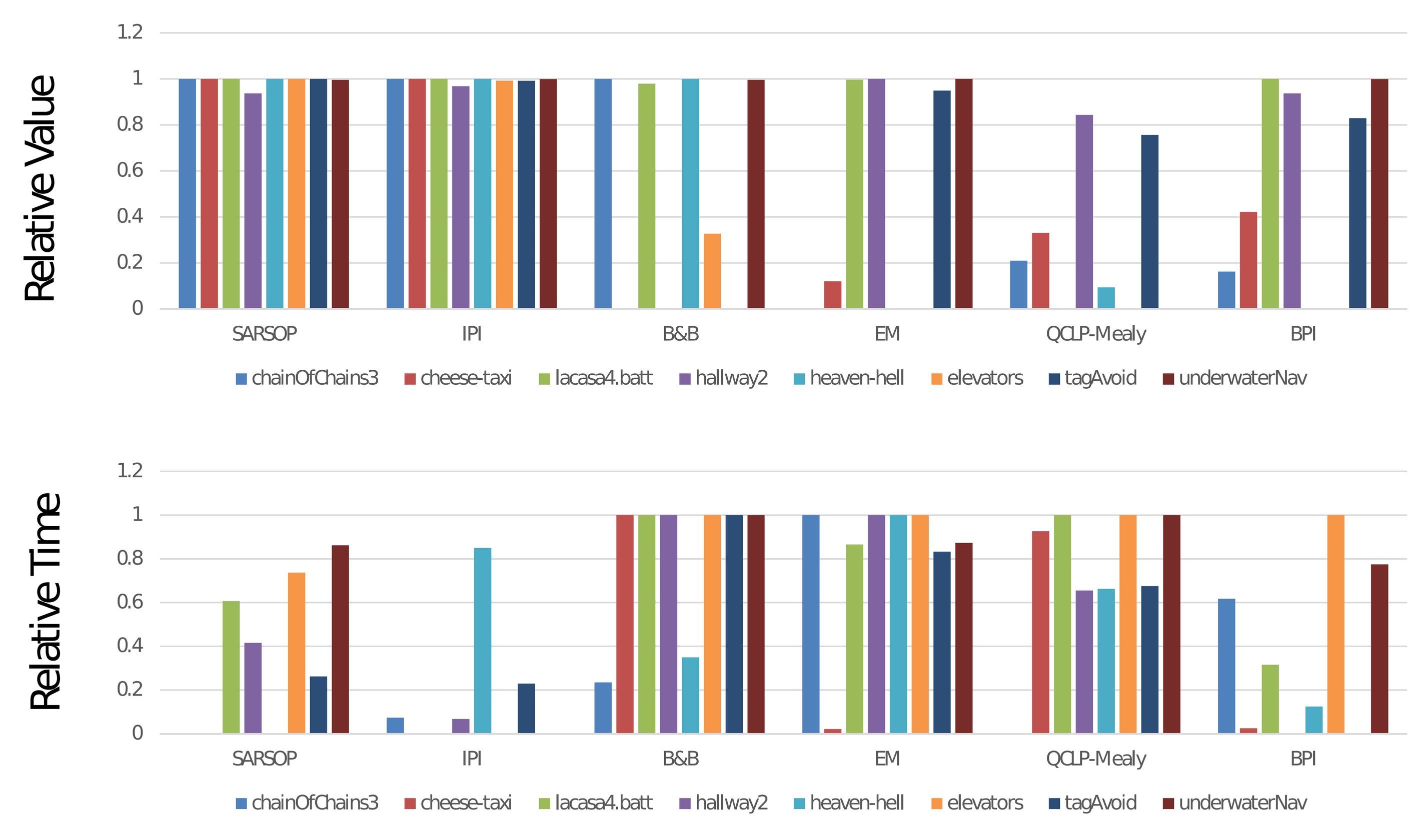


Figure 5 : Relative values: normalised values so that the SARSOP upper bound is 1 and the worst value achieved by any algorithm is 0. Relative time: normalised time where the longest time is 1 and the shortest time is 0.

Conclusion

- A new view on principled methods for policy iteration in POMDPs
- A new efficient method for improving individual nodes
- An intuitive explanation of local optima and challenges in escaping it
- A guaranteed method for escape that facilitates fast, anytime execution
- The best node for escape is deterministic
- Heuristic methods analysed (with new connections identified) and used in a practical and well-justified manner

Acknowledgements

This research was sponsored by NSERC and MITACS.

e-mail: m.grzes@kent.ac.uk and ppoupart@uwaterloo.ca

www: http://www.cs.kent.ac.uk/people/staff/mg483/