Policy Construction for MDPs Represented in Probabilistic PDDL

Boris Lesner and Bruno Zanuttini

GREYC, Université de Caen Basse-Normandie, FRANCE

June 14 2011

Outline

Motivations

PPDDL actions represent compactly Markov Decision Processes.

How to compute optimal infinite horizon discounted policies with PPDDL actions as input ?

The usual way:

- Translate PPDDL into DBNs.
- Use your favorite solver (e.g. SPUDD).

Or, exploit the PPDDL structure directly

- Avoid the cost of translating into DBNs.
- Handles naturally corellated effects.

Compact Action and Value Function Representation

Grounded PPDDL

- Propositional state variables $X = \{x_1, \ldots, x_n\}$
- State space $S = \{0, 1\}^X$

State updates as Basic Effects

- Basic effect: a set of literals b representing changes on a state.
- Like STRIPS effects, applying b to state s gives state $s' = s[b]$ where values of b are forced in s.

Values functions as Algebraic Decision Diagrams

- Compact representation of $\{0, 1\}^n \to \mathbb{R}$ functions
- Efficient operators on functions

PPDDL at a glance

An action a is:

- a precondition: ϕ_a
- an effect: e_a

Effects are recursively defined as:

- x or $\neg x$: forces the value of variable x
- $r \uparrow v$: add reward v
- $\phi \triangleright e$: effect e occurs when ϕ is true
- $e_1 \wedge \cdots \wedge e_k$: all of $e_1, \ldots, e_k$ occurs, e_i's must be consistent
- $p_1 e_1 | \cdots | p_k e_k$: each e_i may occur with probability p_i.

Effect–Reward Distribution

For a state s, a PPDDL effect e defines a probability distribution $D(e, s)$ over basic-effect–reward pairs $\langle b, r \rangle$.

Frameless Action-Value Functions (F-Values)

Frame Assumption: the variables unchanged by an action remains unchanged after taking the action. There are no exogenous effects.

- Assumed by regular action-value functions:

$$Q_V^e(s) = \mathop{\mathbf{E}}_{\langle b,r\rangle \sim D(e,s)} [r + \gamma V(s[b])]$$

- When not assumed, unchanged variables take value as in $s' \in \{0,1\}^X$:

$$F_V^e(s,s') = \mathop{\mathbf{E}}_{\langle b,r\rangle \sim D(e,s)} [r + \gamma V(s'[b])]$$

- Frameless action-values embed the regular ones:

$$Q_V^e(s) = F_V^e(s,s)$$

Why F-Values ?

Allows incremental handling of conjunctive effects

$$e_1 \wedge e_2 \wedge \cdots \wedge e_k$$

PPDDL convention:

- Each e_i modifies different variables, or at least consistently

Incremental conjunctive effect backup:

- Compute $F_V^{e_1}$, make no assumptions on how variables not modified by e_1 change.
- Next, let $V \leftarrow F_V^{e_1}$ and compute $F_V^{e_2}$ accounting for both e_1 and e_2.
- Repeat.

And more. . .

From PPDDL Effects to F-Values

Backup Rules

Given F-Value V' there is a rule for each kind of effect e to compute $F^{e}_{V'}$.

ADD efficiency

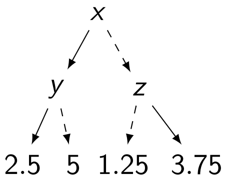
Each rule corresponds to few ADD operations.

Example Action Backup

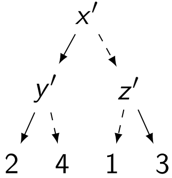
Action effect

$$(r \uparrow 1) \wedge (\neg x \triangleright z) \wedge (0.3 \neg x | 0.7 y)$$

Previous value function V



Primed & γ-discounted F-value V' st. $V'(\cdot, s) = \gamma V(s)$.
$(\gamma = 0.8)$

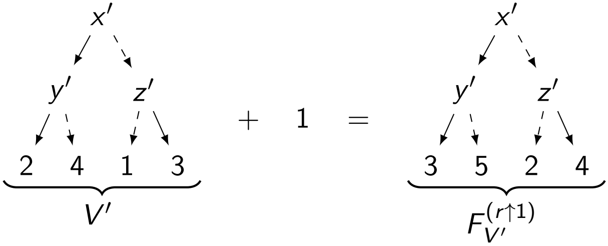


F-Value for an update effect

$$\textcolor{red}{(r \uparrow 1)} \wedge (\neg x \triangleright z) \wedge (0.3 \neg x | 0.7 y)$$

Previous F-value: V'

$$F_{V'}^{(r\uparrow 1)}(s, s') = V'(s, s') + 1$$

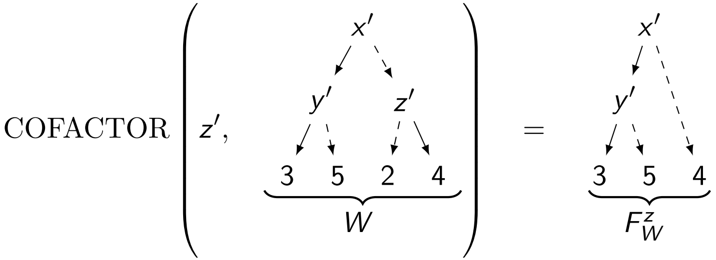


F-Value for a simple effect

$$(r \uparrow 1) \wedge (\neg x \triangleright z) \wedge (0.3 \neg x | 0.7 y)$$

Previous F-value: $W = F_{V'}^{(r \uparrow 1)}$

$$F_W^z(s, s') = W(s, s'[z])$$

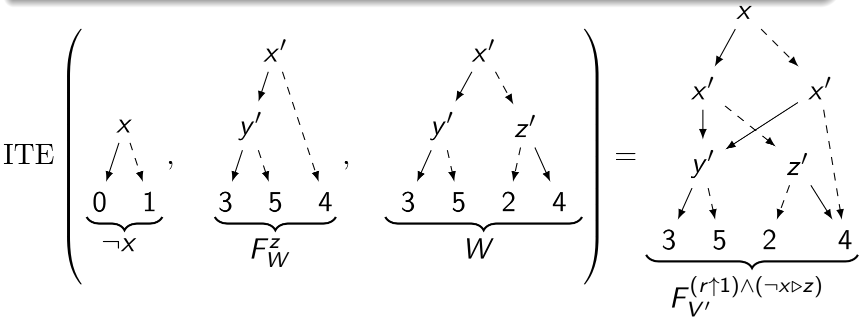


F-Value for a conditional effect

$$(r \uparrow 1) \wedge (\neg x \triangleright z) \wedge (0.3 \neg x | 0.7 y)$$

Previous F-value: $W = F_{V'}^{(r\uparrow 1)}$

$$F_W^{(\neg x \triangleright z)}(s, s') = \begin{cases} F_W^z(s, s') & \text{if } s \models \neg x \\ W(s, s') & \text{otherwise} \end{cases}$$

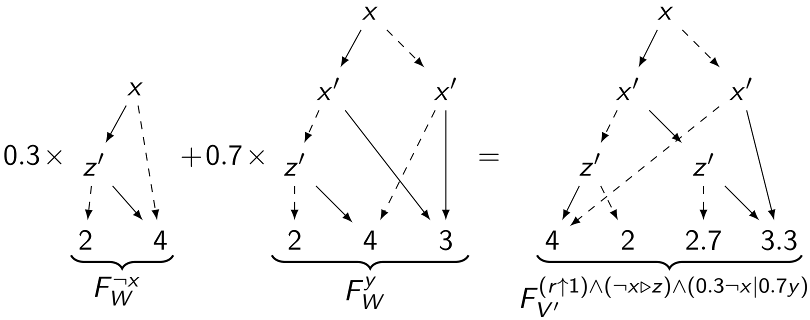


F-Value for a probabilistic effect

$$(r \uparrow 1) \wedge (\neg x \triangleright z) \wedge (0.3 \neg x | 0.7 y)$$

Previous F-value: $W = F_{V'}^{(r\uparrow 1)\wedge(\neg x \triangleright z)}$

$$F_W^{(0.3\neg x|0.7y)}(s, s') = 0.3 \times F_W^{\neg x}(s, s') + 0.7 \times F_W^{y}(s, s')$$

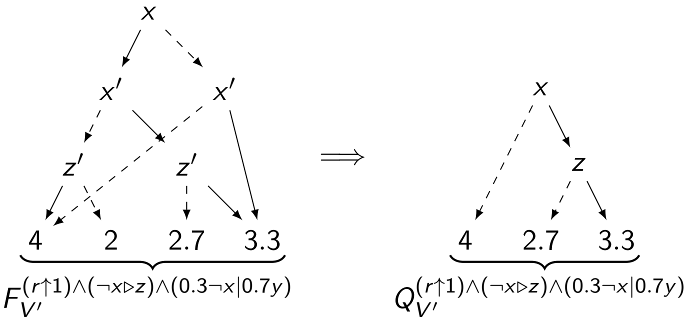


From F-values to action values

$$Q_V^e(s) = F_V^e(s, s)$$

With ADDs:

- "unprime" each primed variable and keep consistent branches.
- or with operators: $Q = \exists X'[x_1 \leftrightarrow x_1' \times \cdots \times x_n \leftrightarrow x_n' 1 \times F]$



Value Iteration with F-Values

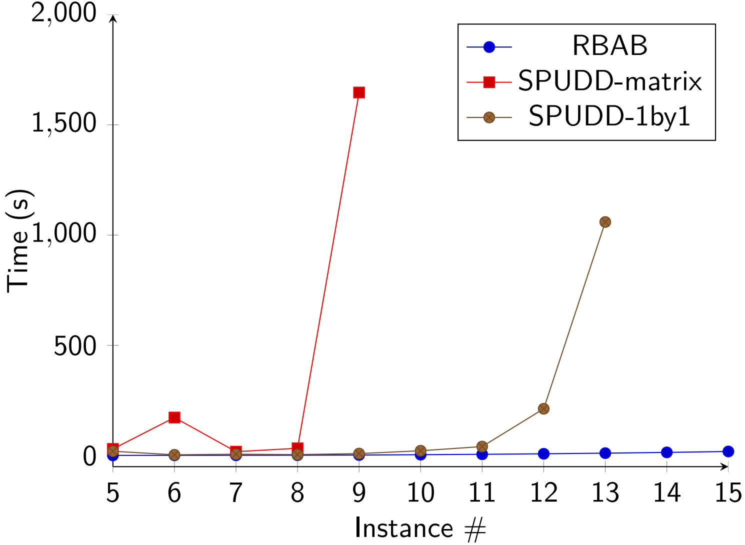
Algorithm: Rule Based Action Backup (RBAB)

A simple adaptation of Value Iteration

- $V \leftarrow 0$
- Repeat until convergence:
  1. $V' \leftarrow \gamma \times \mathrm{PrimeVars}(V)$
  2. Compute $F^{e_a}_{V'}$ for each action a
  3. Deduce $Q^a_{V'}$ from $F^{e_a}_{V'}$
  4. $V \leftarrow \max_a Q^a_{V'}$
- Extract policy

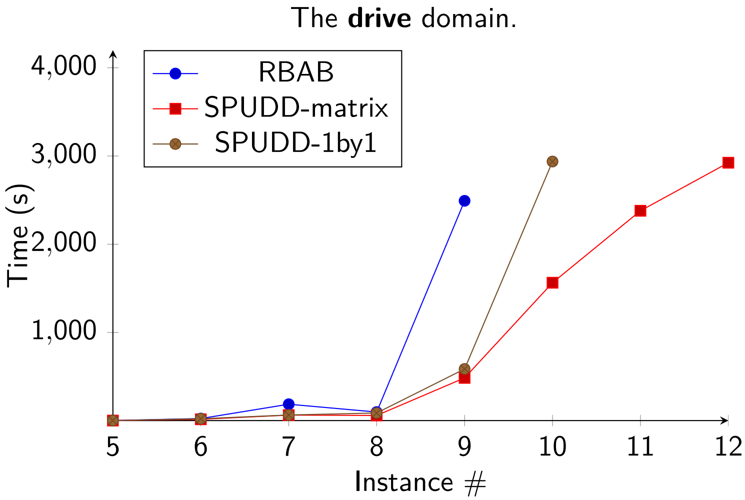
Evaluation on IPC Domains – 1/2

A best-case domain: **search-and-rescue**



Evaluation on IPC Domains – 2/2

Impact of the size of problem description: **drive** domains

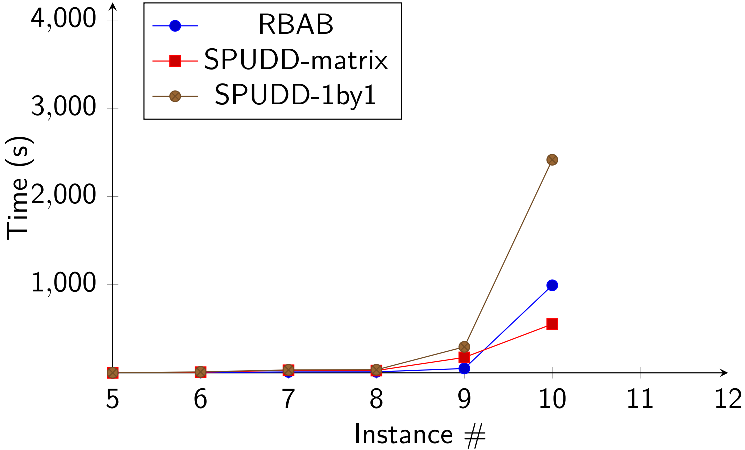


Domain: 3 Action Schemata, Effects: $\bigwedge_i p_i(c_i \triangleright e_i) | p'_i(c'_i \triangleright e'_i)$

Evaluation on IPC Domains – 2/2

Impact of the size of problem description: **drive** domains

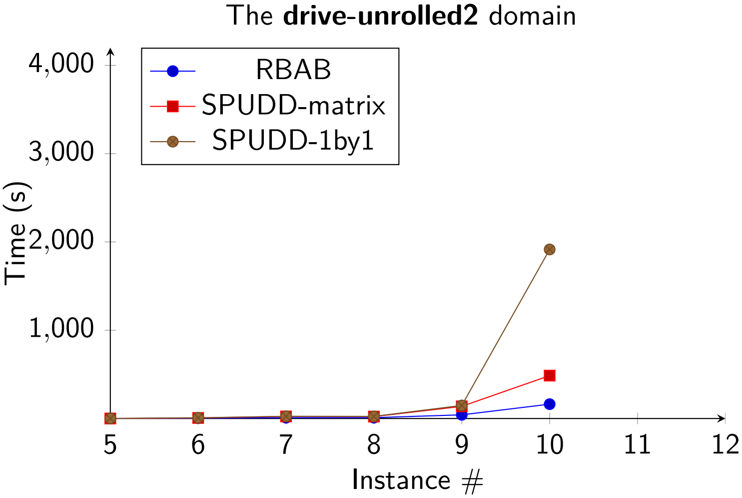
The **drive-unrolled** domain.



Domain: 9 Action Schemata, Effects: $\bigwedge_i p_i(c_i \triangleright e_i) | p'_i(c'_i \triangleright e'_i)$

Evaluation on IPC Domains – 2/2

Impact of the size of problem description: **drive** domains



Domain: 9 Action Schemata, Effects: as compact as possible

A Policy Revision Problem

Revision scenario

Agent has:

- Some policy π and it's value function V.
- A description of an action effect a, and a modified version a'.
- The F-value F_V^a.

Revision problem: compute the F-value $F_V^{a'}$, from F_V^a.

Possible applications: model-based Reinforcement Learning, which incrementally learns action descriptions. Particularly RTDP-RMAX or RTDP-IE which perform one-step action backups.

Possible Revisions

Adding an effect : $a' = a \wedge e$

$\rightsquigarrow F_V^{a'} = F_{F_V^a}^e$.

Modifying rewards : $a' = a \wedge (\phi \triangleright (r \uparrow v))$

$\rightsquigarrow F_V^{a'} = F_V^a + \phi \times v$.

Revising probabilities I :

- From $a = \phi \triangleright (p\ e|(1-p)\ \top)$
- To $a' = \phi \triangleright (q\ e|(1-q)\ \top)$

$\rightsquigarrow F_V^{a'} = \mathrm{ITE}(\phi, (1 - \frac{1-q}{1-p}) \times F_V^{a \wedge e} + \frac{1-q}{1-p} \times F_V^a, F_V^a)$

Revising probabilities II :

- From $a = \phi \triangleright (p\ e|(1-p)\ e')$
- To $a' = \phi \triangleright (q\ e|(1-q)\ e')$
- with e and e' consistent

Conclusion

Frameless Value Functions allows

- Value Iteration from PPDDL MDPs
  - No translation into DBNs.
  - Efficient with compact effects and non exclusive conditions
  - Exploit the efficiency & compactness of ADDs.
- Also offers possibilities for policy revision

Perspectives

- Probabilisting planning (i.e. using initial & goal states)
- Approximate value iteration (like APRICODD)
- Experiment with Affine ADDs.

Conclusion

Frameless Value Functions allows

- Value Iteration from PPDDL MDPs
  - No translation into DBNs.
  - Efficient with compact effects and non exclusive conditions
  - Exploit the efficiency & compactness of ADDs.
- Also offers possibilities for policy revision

Perspectives

- Probabilisting planning (i.e. using initial & goal states)
- Approximate value iteration (like APRICODD)
- Experiment with Affine ADDs.

Thank You.