Symbolic Dynamic Programming for First-order POMDPs

AAAI 2010

Scott Sanner

NICTA

ANU
THE AUSTRALIAN NATIONAL UNIVERSITY

Kristian Kersting

IAIS

Fraunhofer Institut
Intelligente Analyse- und Informationssysteme

Relational Observation Spaces: The POMDP Killer

- World is full of observations:
  - *next-to (Alice, Bob)*
  - *in-front (Bob, door)*
  - *on (coffee, table)*

  $|\Delta| = n$, k binary predicates → $|O| = 2^{kn^2}$

Most POMDPs have carefully defined observation spaces

- DP backup in POMDP solution $\propto |S||\Gamma|^{|O|}$

- **Relevant** observations depend on task
  - Alice wants to exit… $\exists x$ in-front (x, door)
  - Alice wants coffee… $\exists y$ on (coffee, y)

Can we **derive** relevant observations?

Differences from Wang & Khardon (AAAI-10)

- FO-POMDP foundations same as Wang's thesis (2007) and Wang & Khardon (AAAI-10)
  - Just *case notation* vs. their *FODDs*

- Our key contribution is observation model!
  - For fixed action instance, we allow ∞ observations
  - **Contribution: how to derive FOL formulae defining relevant observations?**

Outline

- POMDP Background
  - Standard definition
  - Running example: Tiger-2010
- Relational observations & FO-POMDPs
  - Model
  - Symbolic dynamic programming solution
    - Extension of Boutilier, Reiter, Price (IJCAI-01)
  - Proof-of-concept evaluation
  - Lifted policies?
- Where to from here?

POMDPs

- POMDP is a tuple ‹**S, A, O, T, Z, R**›

- **S:** set of states
- **A:** set of actions
- **O:** set of observations

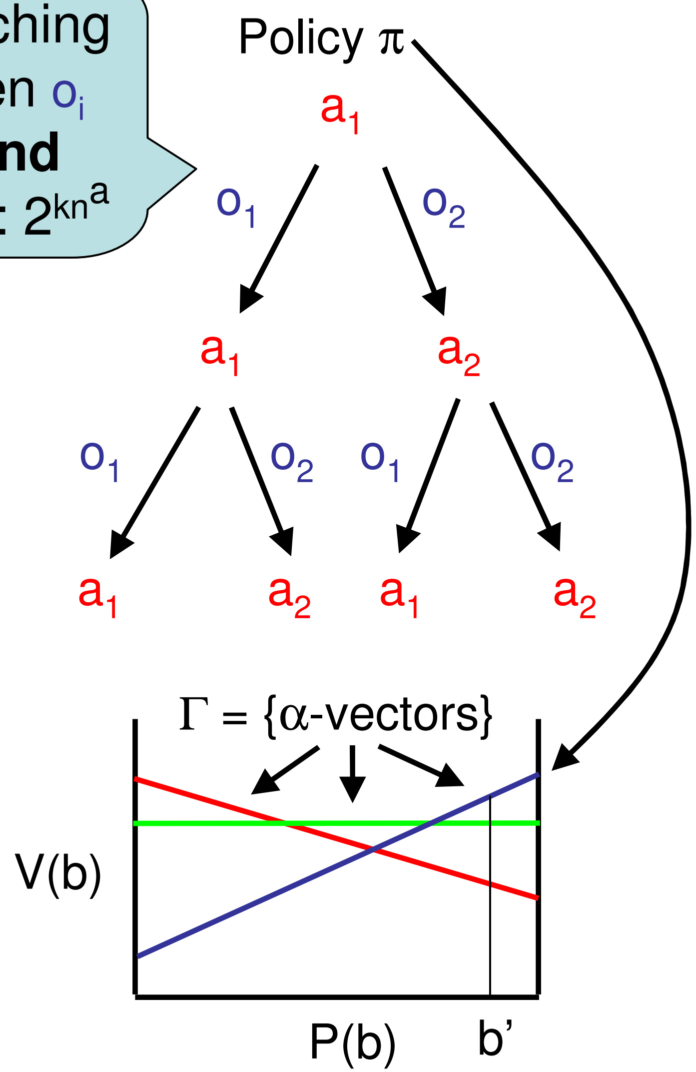
- **$T(s',a,s) := P(s'|s, a)$:** transition function
- **$Z(o,a,s') := P(o|a, s')$:** observation function
- **$R(s,a)$:** reward function

Solving POMDPs

Huge branching factor when o_i are **ground relational**: 2^{kn^a}

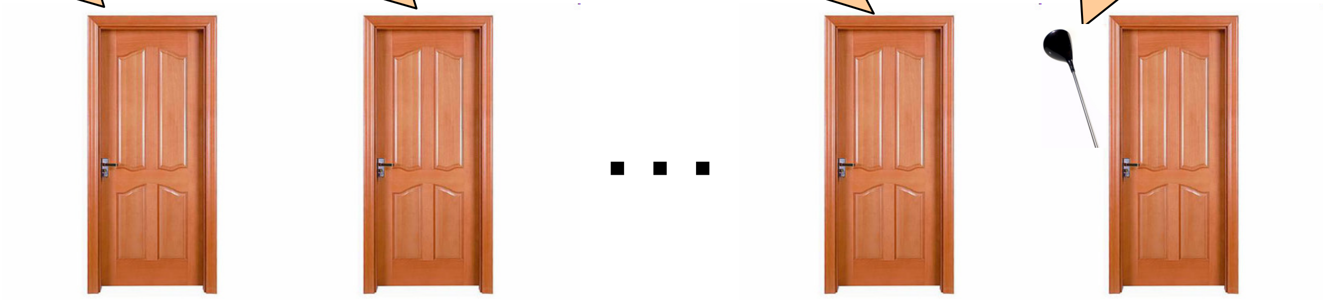
- Maximize expected sum of discounted rewards

- Known solution (Sondik, 1971):
  - Policy π = conditional plan
  - α-vector: linear value function of belief state b for π
  - Value: max over Γ = {α-vectors }
  - In b': use best policy from Γ



FO-POMDP: Tiger-Door 2010

Nothing Nothing Nothing Noise



d_1 d_2 … d_{14} d_{15} …

Could be **large** number of doors, observation for every door!

"Where are you?"

Listens…

Open(d_{15})

$1,000,000,000 loss for non-optimal solution!

Tiger-Door 2010 as an FO-POMDP

- States
  - $\forall i$, wife(d_i) $\in$ {t,f}
- Observations
  - $\forall i$, noise(d_i) $\in$ {t,f}
- Actions
  - listen
  - open(d_i)
- Transition function
  - $\forall i$, wife(d_i): no change
- Reward
  - listen: -1
  - open(d_i):
    - wife(d_i): 10
    - $\neg$wife(d_i): -100
- Observation function…

Observations for Tiger-Door 2010

- open(d_i):
  - Generates: null observation
- listen:
  - Generates: $\forall i$, noise(d_i) $\in$ {t,f}
    - Fails with probability .3 (no noise)
  - Prob. over *deterministic observation action outcomes*
    - P($listenSucc_O$ | listen) = .7
    - P($listenFail_O$ | listen) = .3
  - Define SitCalc effects (S → O)
    - wife(d_i) $\wedge$ a=$listenSucc_O$ $\supset$ noise(d_i)
    - … $\supset$ $\neg$noise(d_i)

Handle observations as implicit **observation actions**

Effect axioms have to be **progressable**

- Reiter, 2001
- Vassos *et al*, 2008

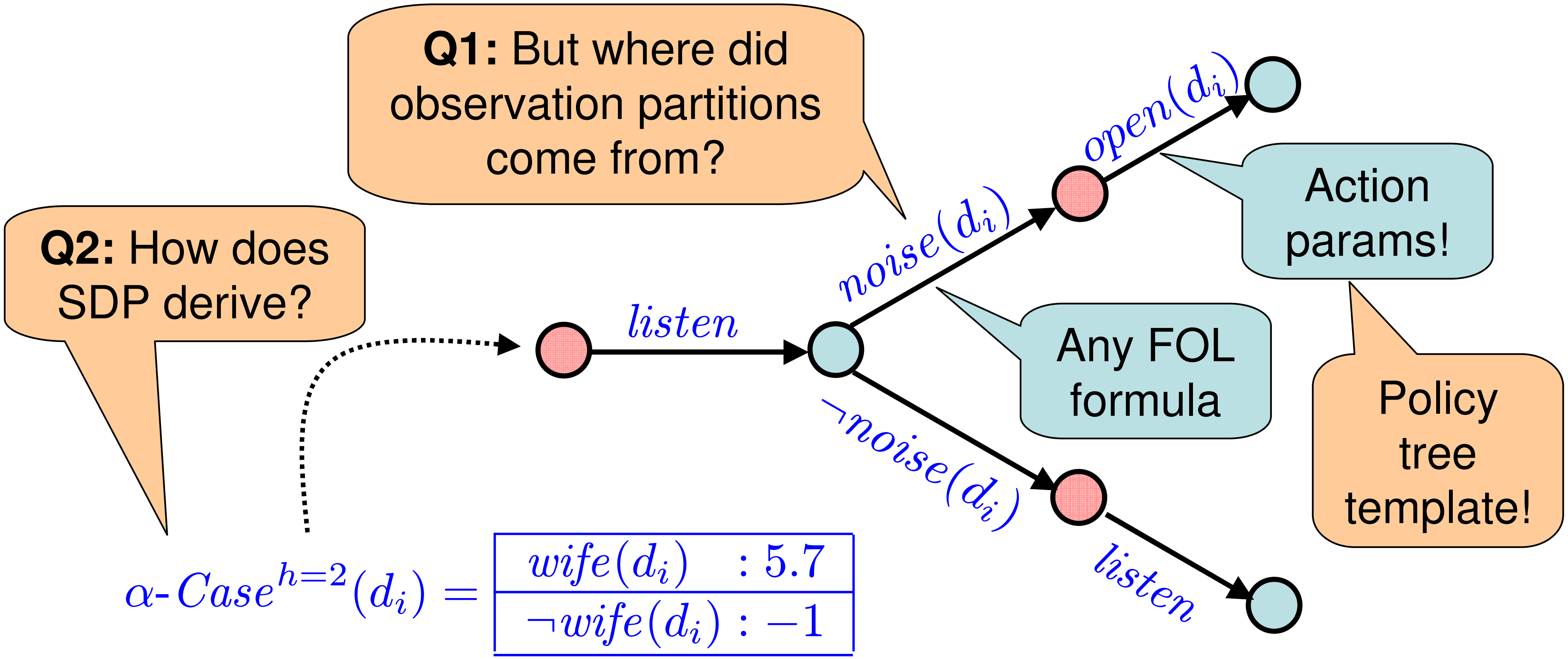
Where are we?

- FO-POMDP model is “defined”
- Now on to the Symbolic Dynamic Programming (SDP) solution…
- Questions:

  (1) What is a policy tree for an FO-POMDP?

  (2) How to compute first-order α-vector?

FO-Policy Trees and α-Case



$$\alpha\text{-}Case^{h=2}(d_i) = \boxed{\begin{array}{l} wife(d_i) \quad : 5.7 \\ \hline \neg wife(d_i) : -1 \end{array}}$$

- Each policy corresponds to an α-Case
  – **Compact** relational α-vector

Q1: Deriving the Observations at Horizon h

- **Relevant States S^h:**
  take cross-product of all Γ^h = {α-Case}

$$\begin{array}{|c|}\hline \phi_1 \\ \hline \phi_2 \\ \hline \phi_3 \\ \hline \end{array} \times \begin{array}{|c|}\hline \phi_1 \\ \hline \neg\phi_1 \\ \hline \end{array} = \begin{array}{|c|}\hline \phi_1 \\ \hline \phi_2 \\ \hline \phi_3 \\ \hline \end{array}$$

- **Relevant Observations O^h**

$$P(o_1|s) \otimes \begin{array}{|c|}\hline Prog(\phi_1, o_1) \\ \hline Prog(\phi_2, o_1) \\ \hline Prog(\phi_3, o_1) \\ \hline \end{array}$$

$$\oplus$$

$$P(o_2|s) \otimes \begin{array}{|c|}\hline Prog(\phi_1, o_2) \\ \hline Prog(\phi_2, o_2) \\ \hline Prog(\phi_3, o_2) \\ \hline \end{array}$$

$s \in S^h$

$$\begin{array}{|l|}\hline Prog(\phi_1, o_1) \wedge Prog(\phi_1, o_2) \wedge \ldots : 0.1 \\ \hline \cdots \qquad\qquad\qquad\qquad : \cdots \\ \hline Prog(\phi_3, o_1) \wedge Prog(\phi_3, o_2) \wedge \ldots : 0.2 \\ \hline \end{array}$$

$P^h(o|s) \longleftarrow o \in O^h$

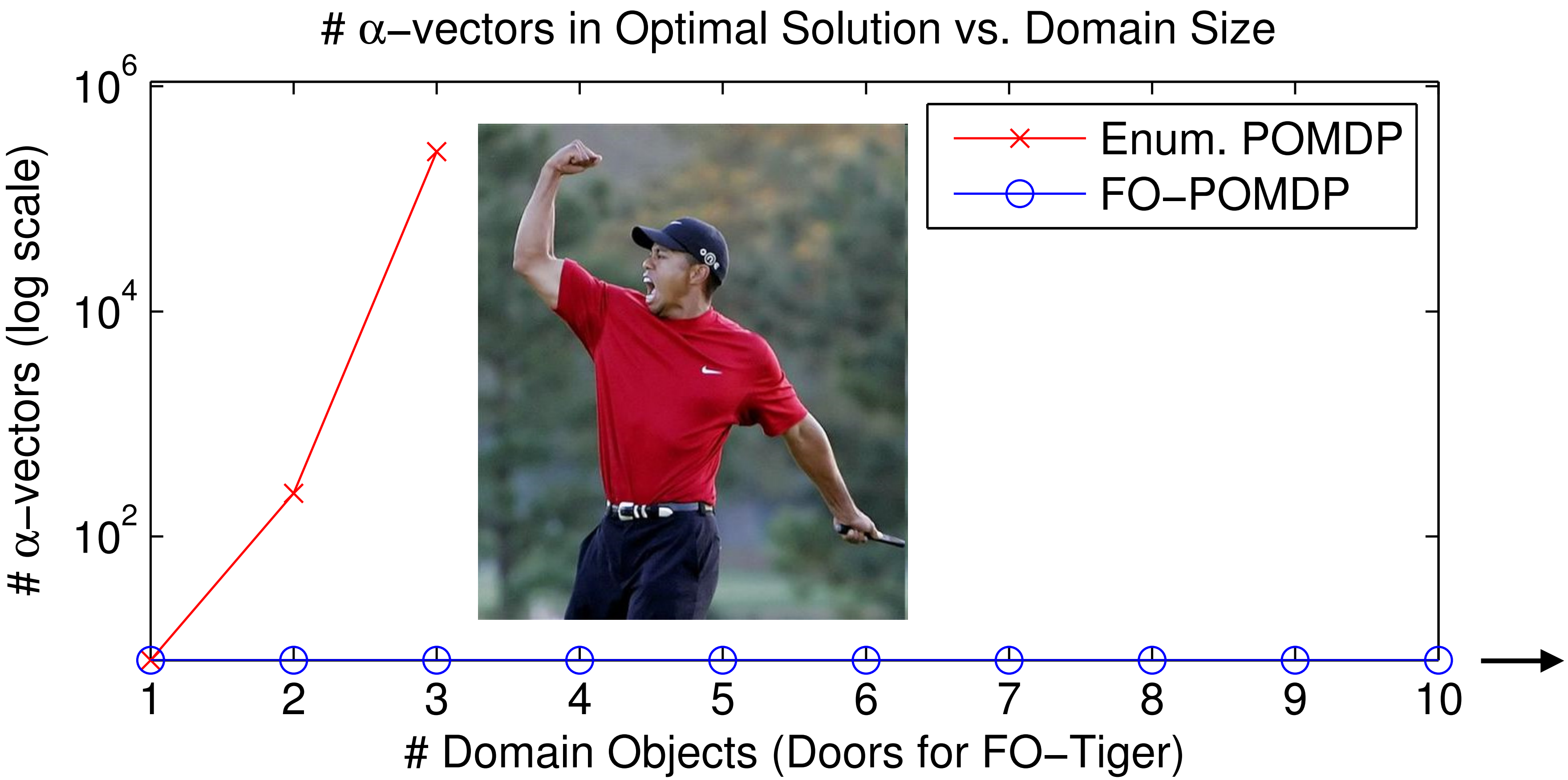
Q2: Symbolic Dynamic Programming (SDP) for FO-POMDPs

- Painful notational details in paper
- Similar to SDP for FO-MDPs
  - But cannot max over action params (Wang & Khardon, AAAI-10)
    - Can do later when dominance pruning
  - When apply SDP at horizon h
    - Derive $P^h(o|s)$ ← case statement over s for each $o \in O^h$
    - $\oplus_o$ Regr($P^h(o|s)$, A) ⊗ [Standard FO-DTR Backup for A]
  - Same as Wang (2007), Wang & Khardon (AAAI-10)
    - Except we must derive $P^h(o|s)$ (in their case, it's fixed)

Boutilier, Reiter, Price (IJCAI-01)

Empirical Comparison

- # of α-vectors vs. α-Cases for FO-Tiger



Future Work I

- Tractability
  - Need more structure than case statements
  - E.g., FODDs (Wang, Joshi, & Khardon, 2008)
    FOADDs (Sanner & Boutilier, 2009)
- Action abstraction
  - **Can do** during α-Case pruning
    - Once you've heard a noise at door(d_i), open d_i
      - dominates all other actions
    - Example in paper, need general procedure

Future Work II

- Observation models
  - Handle non-progressable observation models? Yes!
  - Functional (continuous) fluents: *temperature > 44C*
- Information-gathering FOMDP
  - Middle-ground between FO-MDP and FO-POMDP
  - Captures Tiger-Door 2010 (but not all FO-POMDPs)
- Factored POMDPs
  - Factored observation space!
  - Derive compact policy trees

Summary

- Rich relational observation spaces
  – very natural
  – but kill enumerated POMDP solutions

- We need FO-POMDP algorithms that can **derive** relevant observations

- Gave example of SDP & FO-POMDPs
  – **Rich area for further research**