Fast Probabilistic Planning Through Weighted Model Counting

Carmel Domshlak   Jörg Hoffmann

Technion (Israel)

Max-Planck-Institute for Computer Science (Germany)

June 8, 2006

Talk Outline

- **Probabilistic Planning**
- Probabilistic-FF:
  - Search States as Bayes networks
  - Search States as Weighted CNFs
  - Heuristic Function
- Results
- Conclusion

Probabilistic Planning: Problem Definition

Also known as conformant/conditional probabilistic planning

- Initial (belief) state: probability distribution P_I over the world states
- A set of (possibly) stochastic actions
- Goal: a set of goal world states
- Plan: a single sequence of actions that transforms the system into one of the goal states with probability higher than θ

Probabilistic Planning: Declarative Specification

STRIPS-like, declarative description: (A, P_I, G, θ)

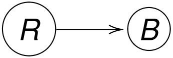
- Initial belief state P_I in structured representation
  - *Bayes network* $\mathcal{N}_I$ over state propositions/variables
- Deterministic actions
  - STRIPS plus conditional effects
- Probabilistic actions
  - STRIPS plus *conditional PDs* over effects
- Goal G: a set of facts

Probabilistic actions: treated by the framework, but yet to be implemented

Example

- Locations L_1, L_2, robot R, block B
- Actions
  - **robot moves between locations** (deterministic)
    - robot in the target location with probability 1
  - **robot moves between locations while carrying the block** (probabilistic)
    - success with probability 0.7
    - robot moves, but block stays with probability 0.2
    - complete failure with probability 0.1
- Initial belief state by $\mathcal{N}_I$:

rL_1	rL_2
0.9	0.1



	bL_1	bL_2
rL_1	0.7	0.3
rL_2	0.2	0.8

Talk Outline

- Probabilistic Planning
- **Probabilistic-FF:**
  - Search States as Bayes networks
  - Search States as Weighted CNFs
  - Heuristic Function
- Results
- Conclusion

Probabilistic-FF: Informal Overview

- ... is based on the Conformant-FF code
- ... "simplifies" to Conformant-FF when $\theta = 1$
- ... extends Conformant-FF's belief state representation and heuristic function
- ... tests on problems with probabilistic initial state and deterministic actions
  - state of the art: $\approx$100 world states, 15-20 steps plans
  - solved problems with *billions* world states, $>$ 120 plan steps

Probabilistic-FF: Key issues

- Key ideas: combining between
  1. lazy CNF-based (non-probabilistic) belief state representation of Conformant-FF [BH04]
  2. probabilistic reasoning using weighted CNF model counting [SBK05]

  - gluing between (1) and (2) with lazy representation of belief states using Bayes networks
    - structured representation based on *logical factoring*

- Most technically involved part
  - proper modification of the **heuristic function** (relaxed plans)

Talk Outline

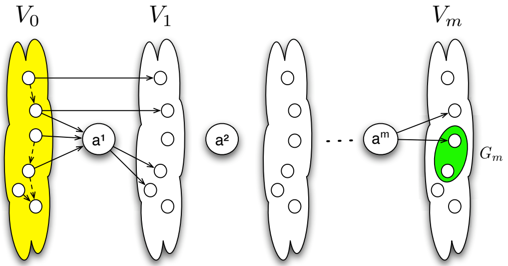
- Probabilistic Planning
- Probabilistic-FF:
  - **Search States as Bayes networks**
  - Search States as Weighted CNFs
  - Heuristic Function
- Results
- Conclusion

Search States as Bayes networks

- Forward search in belief space
  - Search states are belief states reachable from P_I through some actions sequence **a**
- Problem: Explicit belief state description is getting less and less structured with $|\mathbf{a}| \to \infty$

Search States as Bayes networks

- Forward search in belief space
  - Search states are belief states reachable from P_I through some actions sequence **a**
- Problem: Explicit belief state description is getting less and less structured with $|\mathbf{a}| \to \infty$
- Solution: Lazy representation of the belief state "after **a**" as a Bayes network $\mathcal{N}_{\mathbf{a}}$



- $P_{\mathbf{a}} = P(V_m)$
- $P_{\mathbf{a}}(G) = P(G_m)$

Talk Outline

- Probabilistic Planning
- Probabilistic-FF:
  - Search States as Bayes networks
  - **Search States as Weighted CNFs**
  - Heuristic Function
- Results
- Conclusion

Inference in BNs and Weighted CNFs

- Problems:
  - Inference in BNs is #P-complete
  - Classical exact algorithms do not scale well on large, dense networks

Inference in BNs and Weighted CNFs

- Problems:
  - Inference in BNs is #P-complete
  - Classical exact algorithms do not scale well on large, dense networks
- Suggestion: [CD05,SBK05]
  1. Compile a BN $\mathcal{N}$ into a cnf $\varphi(\mathcal{N})$,
  2. Associate some literals of $\varphi(\mathcal{N})$ with numerical *weights* derived from $\mathcal{N}$,
  3. Do *weighted model counting* on $\varphi(\mathcal{N})$ by reusing (and adapting) techniques used in DPLL-style search for SAT.

Inference in BNs and Weighted CNFs

- Problems:
  - Inference in BNs is #P-complete
  - Classical exact algorithms do not scale well on large, dense networks
- Suggestion: [CD05,SBK05]
  1. Compile a BN $\mathcal{N}$ into a cnf $\varphi(\mathcal{N})$,
  2. Associate some literals of $\varphi(\mathcal{N})$ with numerical *weights* derived from $\mathcal{N}$,
  3. Do *weighted model counting* on $\varphi(\mathcal{N})$ by reusing (and adapting) techniques used in DPLL-style search for SAT.
- Scales well when ...
  - ... lots of deterministic dependencies
  - ... lots of context-specific independencies

We have that!

WMC in Probabilistic-FF Forward Search

Sketch

At a (belief) search state "after **a**" ($\mathcal{N}_{\mathbf{a}}$) ...

- compile $\mathcal{N}_{\mathbf{a}}$ into a wcnf $\varphi(\mathcal{N}_{\mathbf{a}})$
- compute $P_{\mathbf{a}}(G) = \text{WMC}\left(\varphi(\mathcal{N}_{\mathbf{a}}) \wedge G_{|\mathbf{a}|}\right)$
- if $P_{\mathbf{a}}(G) \geq \theta$: return **a**
- otherwise:
  - determine actions *a applicable* "after **a**" (that is, $P_{\mathbf{a}}(pre(a)) = 1$)
  - compute heuristic estimates for belief states "after **a** and *a*"
  - keep searching ...

At a (belief) search state "after **a**" ($\mathcal{N}_{\mathbf{a}}$) ...

- compile $\mathcal{N}_{\mathbf{a}}$ into a wcnf $\varphi(\mathcal{N}_{\mathbf{a}})$
  - ... compilation scheme along [SBK05]
- compute $P_{\mathbf{a}}(G) = \text{WMC}\left(\varphi(\mathcal{N}_{\mathbf{a}}) \wedge G_{|\mathbf{a}|}\right)$
  - ... use Cachet [SBBKP04]
- if $P_{\mathbf{a}}(G) \geq \theta$: return **a**
- otherwise:
  - determine actions *a applicable* "after **a**" (that is, $P_{\mathbf{a}}(pre(a)) = 1$)
  - compute heuristic estimates for belief states "after **a** and *a*"
  - keep searching ...

WMC in Probabilistic-FF Forward Search

Some implementation details

At a (belief) search state "after **a**" ($\mathcal{N}_{\mathbf{a}}$) ...

- compile $\mathcal{N}_{\mathbf{a}}$ into a wcnf $\varphi(\mathcal{N}_{\mathbf{a}})$
  - ... compilation scheme along [SBK05]
- compute $P_{\mathbf{a}}(G) = \text{WMC}\left(\varphi(\mathcal{N}_{\mathbf{a}}) \wedge G_{|\mathbf{a}|}\right)$
  - ... use Cachet [SBBKP04]
- if $P_{\mathbf{a}}(G) \geq \theta$: return **a**
- otherwise:
  - determine actions *a applicable* "after **a**" (that is, $P_{\mathbf{a}}(pre(a)) = 1$)
    - ... as in Conformant-FF: SAT queries only
  - compute heuristic estimates for belief states "after **a** and *a*"
    - ... see next
  - keep searching ...

Talk Outline

- Probabilistic Planning
- Probabilistic-FF:
  - Search States as Bayes networks
  - Search States as Weighted CNFs
  - **Heuristic Function**
- Results
- Conclusion

Probabilistic-FF: Heuristic Function

Heuristic in Conformant-FF:

- . . . ignore delete lists
- . . . ignore all but one effect condition
- . . . extend FF's relaxed planning graph with sets of *unknown* (uncertain) facts, and implications between them
- With unitary effect conditions, the implications are *edges in a DAG* $\Rightarrow$ (practically) efficient reasoning possible

Probabilistic-FF: Heuristic Function
VERY Sketchy

Idea I: Certain *weighted extension* of the implication graph

- No changes in implications due to deterministic actions
  - Implication $c(t) \rightarrow q(t+1)$ for an unknown condition c of (an effect e of) $a \in A(t)$ such that $q \in add(e)$
- Probabilistic actions
  - Special weighted propositions $w_q^\epsilon(t)$ for probabilistic outcomes ϵ of e ($weight(w_q^\epsilon) = prob(\epsilon)$)
  - Implication $w_q^\epsilon(t) \rightarrow q(t+1)$
  - Implication $c(t) \rightarrow w_q^\epsilon(t)$ for an unknown condition $c \ldots$

Idea II: Weight propagation to the leafs of the implication graph

- For *each* fact node $q(t)$, compute $weight_{q(t)}(v)$ for all nodes v in the implication sub-graph $Imp_{q(t)}$ rooted in $q(t)$
- Computed inductively from $q(t)$ down to the leafs of $Imp_{q(t)}$

- $weight_{q(t)}(v)$ is an upper bound on the probability of achieving q at time t by a sequence of actions responsible for a path from v to $q(t)$ in Imp
- The likelihood of achieving the goals by a relaxed plan of t time steps is estimated by:

$$prob(G, t) = \text{WMC}(\varphi(\mathcal{N}_I) \wedge \bigwedge_{g \in G} \bigvee_{leaf \in Imp_{g(t)}} leaf)$$

Probabilistic-FF: Heuristic Function
VERY Sketchy

- Build pRPG until either levels off **or** $prob(G, t) \geq \theta$
- If $prob(G, t) < \theta$: report FALSE
- Otherwise: extract a relaxed plan, and return the number of its actions as *h*

- Completeness: if FALSE, then there is no relaxed plan that achieves *G* with probability $\geq \theta$

-