On representing planning domains under uncertainty

Felipe Meneguzzi – CMU
Yuqing Tang – CUNY
Simon Parsons – CUNY
Katia Sycara - CMU
Outline

• Planning
  – Markov Decision Processes
  – Hierarchical Task Networks
• States and State-Space
• Using HTNs to represent MDPs
• Increasing Efficiency
• Future Work
• Conclusions
Planning

• Planning algorithms more or less divided into:
  – Deterministic
  – Probabilistic

• Formalisms differ significantly
  – Domain representation
  – Concept of solution
    • Plan
    • Policy
Blogohar Scenario

• Original scenario consists of two players planning for concurrent goals
  – NGO
  – Military

• Here, we consider a (simplified) planning task for the military planner
  – Select forces to attack militant strongholds
  – Move forces to strongholds and attacking
Hierarchical Task Networks

- Offshoot of classical planning
- Domain representation more intuitive to human planners
  - Actions (state modification operators)
  - Tasks (goals and subgoals)
  - Methods (recipes for refining tasks)
- Problem comprises
  - Initial State
  - Task
HTN Domain – Actions

- attack(Vehicle, Target)
- move(Vehicle, From, To, Road)
HTN Methods

• Defeat Insurgents at Stronghold $A$ $t^{DI(T)}$
  – Precondition: Target = $A$
  – Task to decompose: $\text{defeatInsurgents}(A)$
  – Tasks replacing $\text{defeatInsurgents}(A)$:
    • $\text{attackWithHumvee}(A)$
    • $\text{attackWithAPC}(A)$
• Attack T with *Humvee*

  – Precondition: $\text{vehicle} (\text{humvee}, V) \land \neg \text{committed} (V)$
  – Task to decompose: $t^{AHu(T)}$
  – Tasks replacing $\text{attackWithHumvee}(T)$:
    • $\text{move}(V, T)$
    • $\text{attack}(V, T)$ – this is an action
• Attack T with APC
  – Precondition:  $\text{vehicle}(\text{apc}, V) \land \neg \text{committed}(V)$
  – Task to decompose: $t^{\text{AA}(T)}$
  – Tasks replacing attackWithAPC(T):
    • move(V,T)
    • attack(V,T) – this is an action
• Move (Route 1)
  – Precondition: Target = A
  – Task to decompose: move(V,T)
  – Tasks replacing move(V,T):
    • move(V,base,tersa,nr1) – These are basic moves
    • move(V,tersa,haram,nr2)
    • move(V,haram,a,sr2)
HTN Methods

• Move (Route 2)
  – Precondition: Target = A
  – Task to decompose: move(\(V,T\))
  – Tasks replacing move(\(V,T\)):
    • move(\(V,\text{base},\text{haram},sr1\)) – These are basic moves
    • move(\(V,\text{haram},a,sr2\))
Methods Summary

• Defeat Insurgents

• Attack with Humvee

• Attack with APC

• Move (Route 1)

• Move (Route 2)

\[ m^{DI(T)} = \begin{cases} T = a, t^{DI(T)}, \{ t^{AHu(T)}, t^{AA(T)} \}, \{ t^{AHu(T)} < t^{AA(T)} \} \end{cases} \]

\[ m^{AHu(T)} = \begin{cases} \text{vehicle(humvee,} V\text{) } \land \neg \text{committed(} V\text{)}, \{ t^{AHu(T)}, \{ t^{Mv(V,} T\text{)}, t^{a(V,} T\text{)} \}, \{ t^{Mv(V,} T\text{)} < t^{a(V,} T\text{)} \} \end{cases} \]

\[ m^{AA(T)} = \begin{cases} \text{vehicle(apc,} V\text{) } \land \neg \text{committed(} V\text{)}, \{ t^{AA(T)}, \{ t^{Mv(V,} T\text{)}, t^{a(V,} T\text{)} \}, \{ t^{Mv(V,} T\text{)} < t^{a(V,} T\text{)} \} \end{cases} \]

\[ T = a, t^{Mv(V,} T\text{)} , \begin{cases} \{ t^{mv(V,} \text{base,} tersa, \text{nr1)} , t^{mv(V,} \text{tersa,} haram, \text{nr2)} , t^{mv(V,} \text{haram,} a, \text{sr2)} \}, , \{ t^{mv(V,} \text{base,} tersa, \text{nr1)} < t^{mv(V,} \text{tersa,} haram, \text{nr2)} < t^{mv(V,} \text{haram,} a, \text{sr2)} \} \end{cases} \]

\[ T = a, t^{Mv(V,} T\text{)} , \begin{cases} \{ t^{mv(V,} \text{base,} haram, \text{hw)} , t^{mv(V,} \text{tersa,} a, \text{sr2)} \}, \{ t^{mv(V,} \text{base,} haram, \text{hw)} < t^{mv(V,} \text{tersa,} a, \text{sr2)} \} \end{cases} \]
• How to execute task defeatInsurgents(a)

\[ t^{DI(T)} \]

– Decompose task through the methods in the domain until actions reached
– Ordered actions is the solution
Decomposed Problem

\[ \text{DI(a)} \]

\[ \text{AHu(a)} \]

\[ \text{AA(a)} \]

\[ \text{Mv(V, T)} \]

\[ \text{a(V, T)} \]

\[ \text{mv(V, base, haram, hw)} \]

\[ \text{mv(V, haram, a, sr2)} \]

\[ \text{mv(V, F, T, R)} \]

\[ \text{a(V, T)} \]

\[ \text{mv(V, haram, a, sr2)} \]
HTN Solution

\[ t_{Dl(a)} \]

\[ t_{AHu(a)} \]

\[ t_{AA(a)} \]

\[ t_{MV(V,T)} \]

\[ t_{A(V,T)} \]

\[ t_{MV(V,T)} \]

\[ t_{A(V,T)} \]

\[ t_{mv(V,base,haram,hw)} \]

\[ t_{mv(V,haram,a,sr2)} \]

\[ a^{mv(V,F,T,R)} \]

\[ a^{mv(V,F,T,R)} \]

\[ a^{mv(V,F,T,R)} \]

\[ a^{mv(V,F,T,R)} \]

\[ a^{mv(V,F,T,R)} \]
Markov Decision Processes

• Mathematical model for decision-making in a partially controllable environment
• Domain is represented as a tuple

\[ \Sigma = (S, A, P) \]

where:
– S is the entire state space
– A is the set of available actions
– P is a state transition function
MDP Domain

- Represented as a hypergraph
- Connections are not necessarily structured
- **All** reachable states are represented
- State transition function specifies how actions relate states
Computing an MDP policy

• An MDP policy is computed using the notion of expected value of a state:

\[
V^*(s) = \max_{a \in A(s)} \left[ u(a,s) + \sum_{s' \in S} \Pr(s'|s)V^*(s') \right]
\]

• Expected value comes from a reward function

• An optimal policy is a policy that maximizes the expected value of every state

\[
\pi^*(s) = \arg \max_{a \in A(s)} \left[ u(a,s) + \sum_{s' \in S} \Pr(s'|s)V^*(s') \right]
\]
MDP Solution

- Solution for an MDP is a policy
- Policy associates an *optimal* action to every state
- Instead of a sequential plan, policy provides contingencies for every state

\[
\begin{align*}
\text{state0} & \rightarrow \text{actionB} \\
\text{state1} & \rightarrow \text{actionD} \\
\text{state2} & \rightarrow \text{actionA}
\end{align*}
\]
States

Hierarchical Task Network

• Not enumerated exhaustively
• State consists of properties of the environment
  \( \text{vehicle(humvee,h1)} \land \text{vehicle(apc,a2)} \)
• Each action modifies properties of the environment
• Set of properties induces a very large state space

Markov Decision Process

• MDP domain explicitly enumerates all relevant states
• Formally speaking, MDP states are monolithic entities
• Implicitly represent the same properties expressed in HTN state
• Large state-spaces make the algorithm flounder
Hierarchical Task Network
- Set of actions induces a smaller state space (still quite large)
- Set of methods induces a smaller still state space
- HTN planning consults this latter state space

Markov Decision Process
- MDP solver must consult the entire state space
- State-space reduction techniques include:
  - Factorization
  - $\epsilon$-homogeneous aggregation
HTNs to represent MDPs

• We propose using HTNs to represent MDPs

• Advantages are twofold:
  – HTNs are more intuitive to SMEs
  – Resulting MDP state-space can be reduced using HTN methods as heuristic
Reachable states
Conversion in a nutshell

• State-space comes from the reachable primitive actions induced my HTN methods

• Probabilities are uniformly distributed over a planner’s choice

• Reward function can be computed using the target states at the end of a plan (Simari’s approach)
Reachable States

\[ s_0 \quad a^{\text{mv}(V,T)} \quad a^{\text{mv}(V,T)} \quad a^{\text{mv}(V,T)} \quad a^{\text{mv}(V,T)} \quad a^{\text{mv}(V,T)} \quad a^{\text{mv}(V,T)} \]

\[ a^{\text{mv}(V,F,T,R)} \quad a^{\text{mv}(V,F,T,R)} \quad a^{\text{mv}(V,F,T,R)} \quad a^{\text{mv}(V,F,T,R)} \quad a^{\text{mv}(V,F,T,R)} \quad a^{\text{mv}(V,F,T,R)} \]

\[ a^{\text{mv}(V,F,T,R)} \quad a^{\text{mv}(V,F,T,R)} \quad a^{\text{mv}(V,F,T,R)} \quad a^{\text{mv}(V,F,T,R)} \quad a^{\text{mv}(V,F,T,R)} \quad a^{\text{mv}(V,F,T,R)} \]

\[ a^{\text{mv}(V,F,T,R)} \quad a^{\text{mv}(V,F,T,R)} \quad a^{\text{mv}(V,F,T,R)} \quad a^{\text{mv}(V,F,T,R)} \quad a^{\text{mv}(V,F,T,R)} \quad a^{\text{mv}(V,F,T,R)} \]

\[ a^{\text{mv}(V,F,T,R)} \quad a^{\text{mv}(V,F,T,R)} \quad a^{\text{mv}(V,F,T,R)} \quad a^{\text{mv}(V,F,T,R)} \quad a^{\text{mv}(V,F,T,R)} \quad a^{\text{mv}(V,F,T,R)} \]

\[ a^{\text{mv}(V,F,T,R)} \quad a^{\text{mv}(V,F,T,R)} \quad a^{\text{mv}(V,F,T,R)} \quad a^{\text{mv}(V,F,T,R)} \quad a^{\text{mv}(V,F,T,R)} \quad a^{\text{mv}(V,F,T,R)} \]
## Conversion example

<table>
<thead>
<tr>
<th>t0</th>
<th>t1</th>
<th>t2</th>
<th>t3</th>
<th>t4</th>
<th>t5</th>
<th>t6</th>
<th>t7</th>
<th>t8</th>
<th>t9</th>
<th>t10</th>
<th>t11</th>
<th>t12</th>
<th>t13</th>
<th>t14</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Increasing Efficiency

• State aggregation using Binary Decision Diagrams (BDDs):
  – BDDs are a compact way of representing multiple logic properties
  – One BDD can represent multiple (factored) states
Limitations and Future Work

• Limitations
  – Current conversion models only uncertainty from the human planner
  – Probabilities uniformly distributed among choices

• Future Work
  – Evaluate quality of compression through $\epsilon$-homogeneity
  – Compute probabilities from the world
Conclusions

• Planning in coalitions is important
• Automated tools for planning need to have a representation amenable to SME
• Our technique offers advantages over either one of the the single approaches:
  – Representation using HTNs for SMEs
  – Underlying stochastic model for military planning using MDPs
QUESTIONS?