On representing planning domains under uncertainty

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

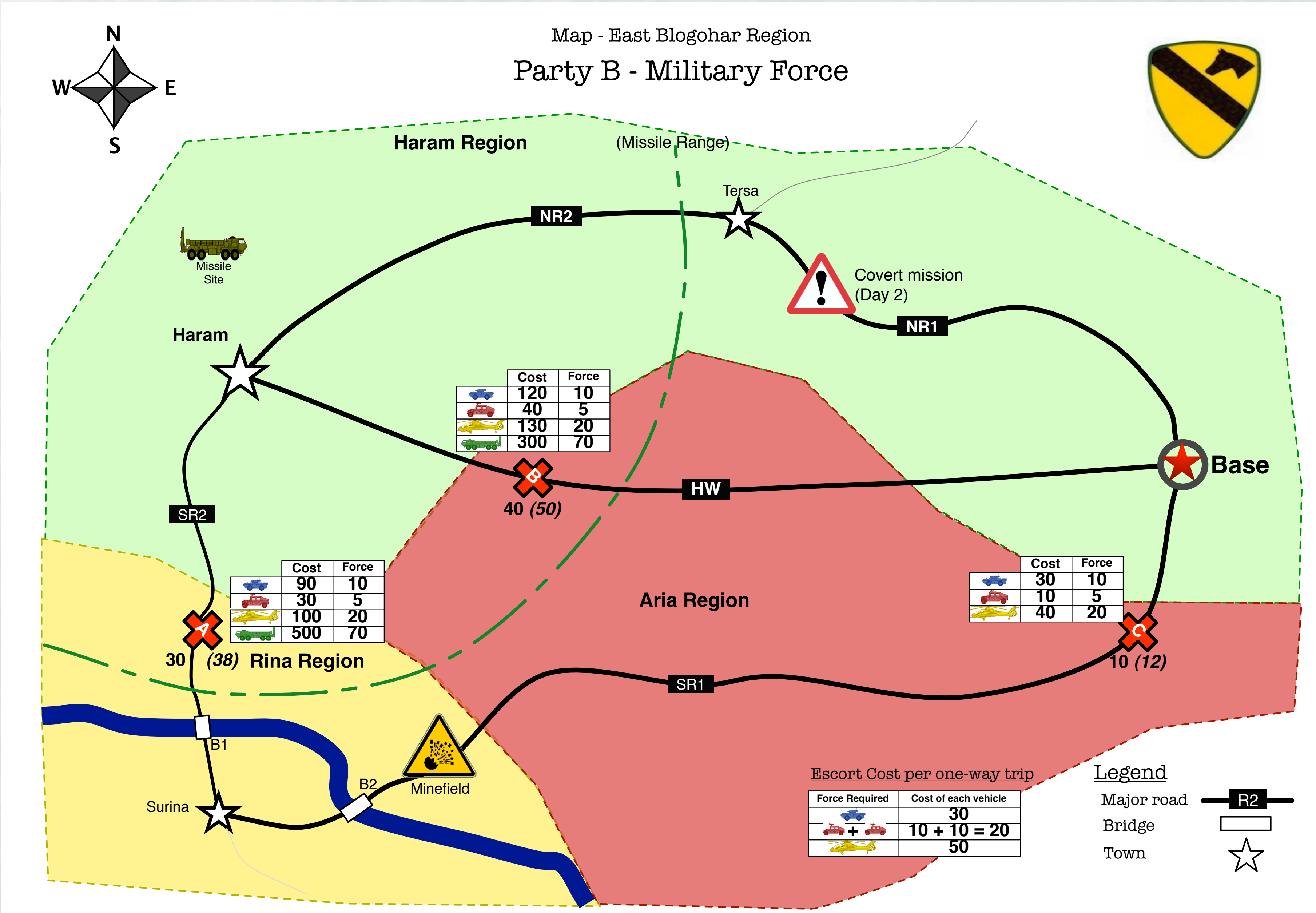
Outline

- 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 (Burnett)



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) $a^{a(V,T)}$
- move(Vehicle, From, To, Road) $a^{mv(V,F,T,R)}$

HTN Methods

- Defeat Insurgents at Stronghold *A* $t^{DI(T)}$
  - Precondition: Target = *A*
  - Task to decompose: *defeatInsurgents(A)*
  - Tasks replacing *defeatInsurgents(A):*
    - attackWithHumvee(A)
    - attackWithAPC(A)

HTN Methods

- Attack T with *Humvee* $t^{AHu(T)}$
  - Precondition: $vehicle(humvee, V) \wedge \neg committed(V)$
  - Task to decompose: attackWithHumvee(T)
  - Tasks replacing attackWithHumvee(T)*:*
    - move(V,T)
    - attack(V,T) – this is an action

HTN Methods

- Attack T with *APC* $t^{AA(T)}$
  - Precondition: $vehicle(apc, V) \wedge \neg committed(V)$
  - Task to decompose: attackWithAPC(T)
  - Tasks replacing attackWithAPC(T)*:*
    - move(V,T)
    - attack(V,T) – this is an action

HTN Methods

- Move (Route 1) $t^{Mv(V,T)}$
  - 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) $t^{Mv(V,T)}$
  - Precondition: *Target* = *A*
  - Task to decompose: move(V,T)
  - Tasks replacing move(V,T)*:*
    - move(V,base,haram,sr1) –These are basic moves
    - move(V,haram,a,sr2)

Methods Summary

- Defeat Insurgents $m^{DI(T)} = \left(T = a, t^{DI(T)}, \left\{t^{AHu(T)}, t^{AA(T)}\right\}, \left\{t^{AHu(T)} \prec t^{AA(T)}\right\}\right)$
- Attack with Humvee

$$m^{AHu(T)} = \begin{pmatrix} vehicle(humvee, V) \wedge \neg committed(V), \\ t^{AHu(T)}, \left\{t^{Mv(V,T)}, t^{a(V,T)}\right\}, \left\{t^{Mv(V,T)} \prec t^{a(V,T)}\right\} \end{pmatrix}$$

- Attack with APC

$$m^{AA(T)} = \begin{pmatrix} vehicle(apc, V) \wedge \neg committed(V), \\ t^{AA(T)}, \left\{t^{Mv(V,T)}, t^{a(V,T)}\right\}, \left\{t^{Mv(V,T)} \prec t^{a(V,T)}\right\} \end{pmatrix}$$

- Move (Route 1)

$$m_1^{Mv(V,T)} = \begin{pmatrix} T = a, t^{Mv(V,T)}, \\ \left\{t^{mv(V,base,tersa,nr1)}, t^{mv(V,tersa,haram,nr2)}, t^{mv(V,haram,a,sr2)}\right\}, \\ \left\{t^{mv(V,base,tersa,nr1)} \prec t^{mv(V,tersa,haram,nr2)} \prec t^{mv(V,haram,a,sr2)}\right\} \end{pmatrix}$$

- Move (Route 2)

$$m_2^{Mv(V,T)} = \begin{pmatrix} T = a, t^{Mv(V,T)}, \\ \left\{t^{mv(V,base,haram,hw)}, t^{mv(V,tersa,a,sr2)}\right\}, \\ \left\{t^{mv(V,base,haram,hw)} \prec t^{mv(V,tersa,a,sr2)}\right\} \end{pmatrix}$$

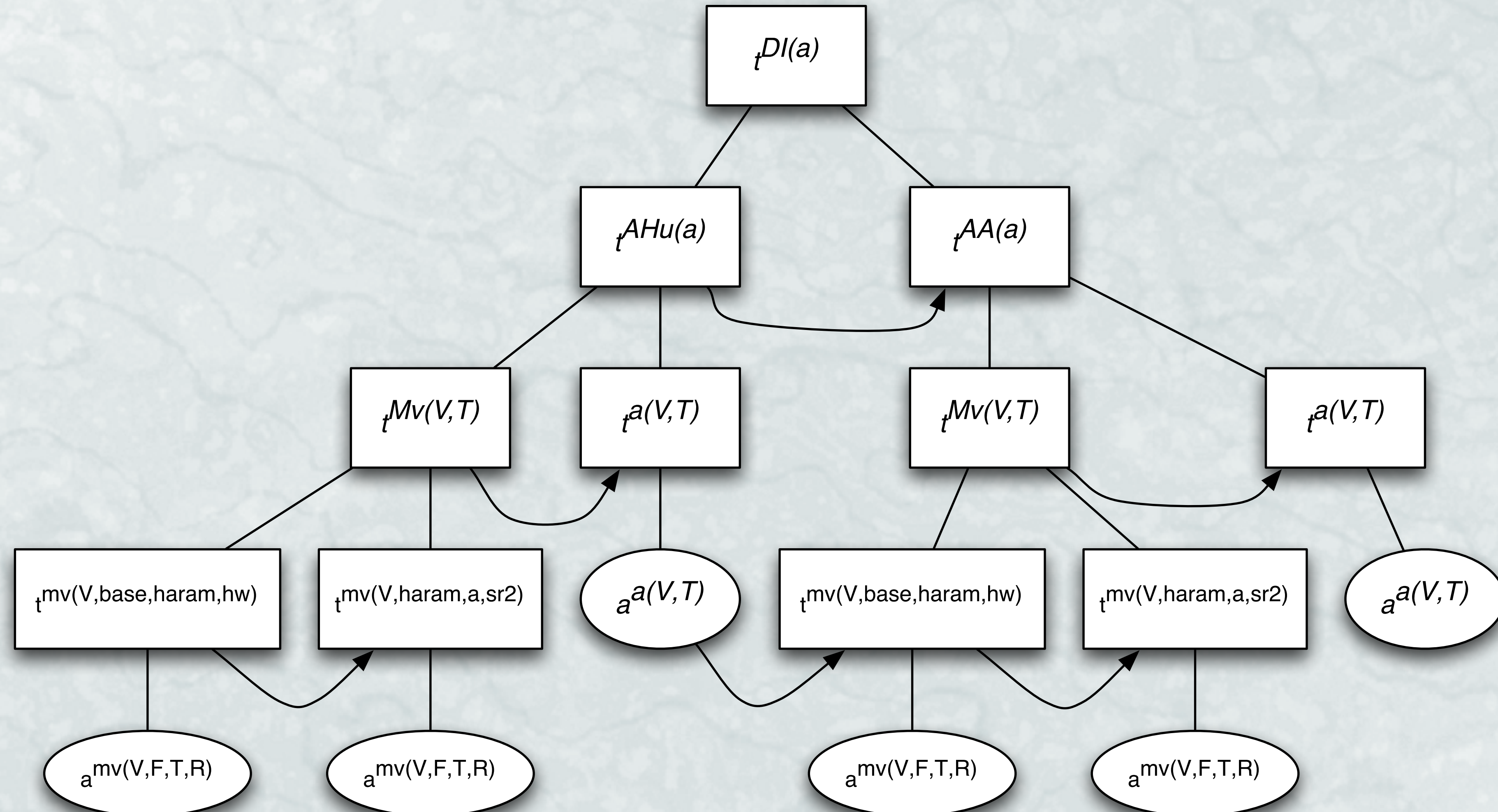
HTN Problem

- How to execute task defeatInsurgents(a)

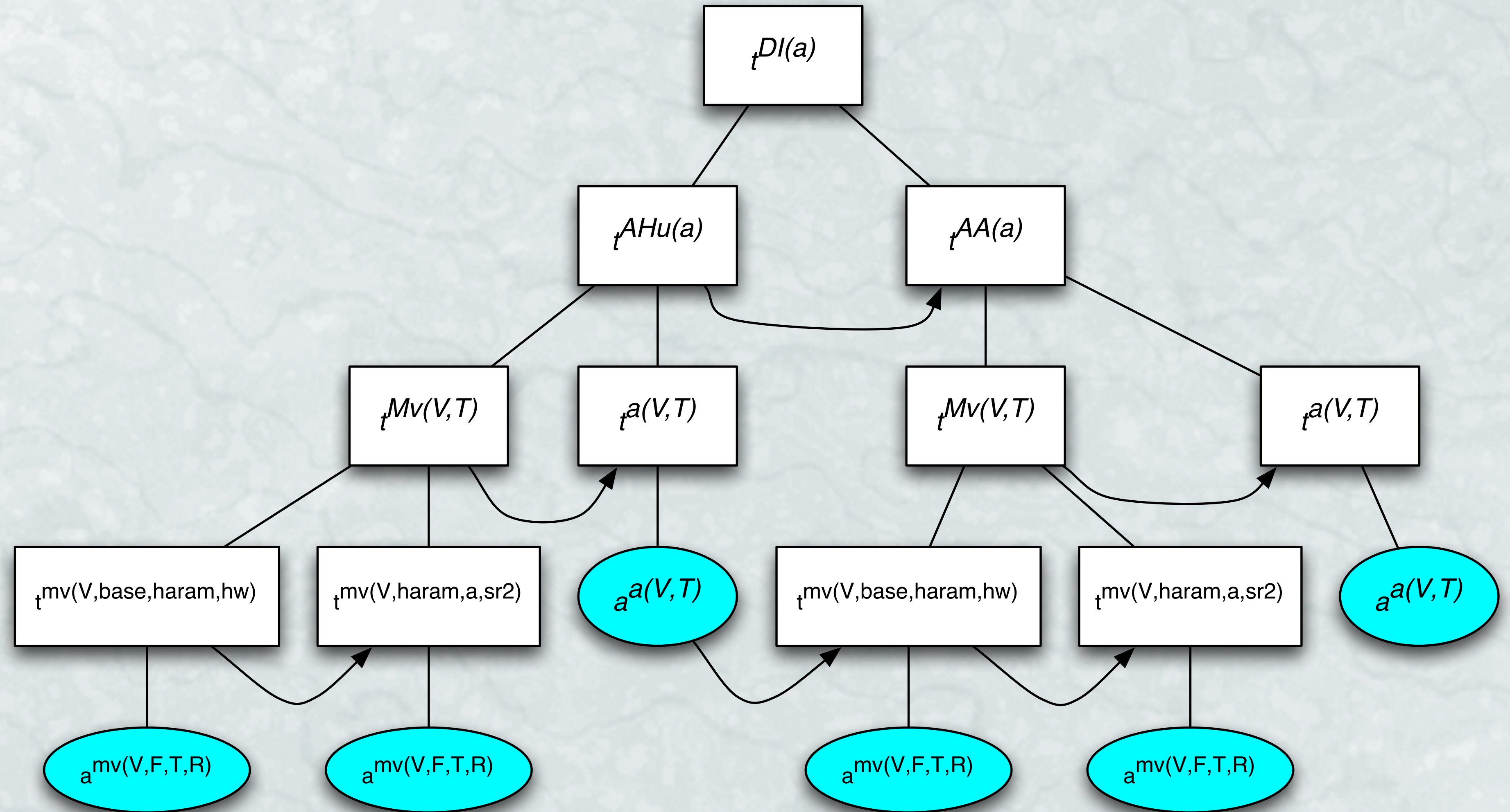
$$t^{DI(T)}$$

  - Decompose task through the methods in the domain until actions reached
  - Ordered actions is the solution

Decomposed Problem



HTN Solution



Markov Decision Processes

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

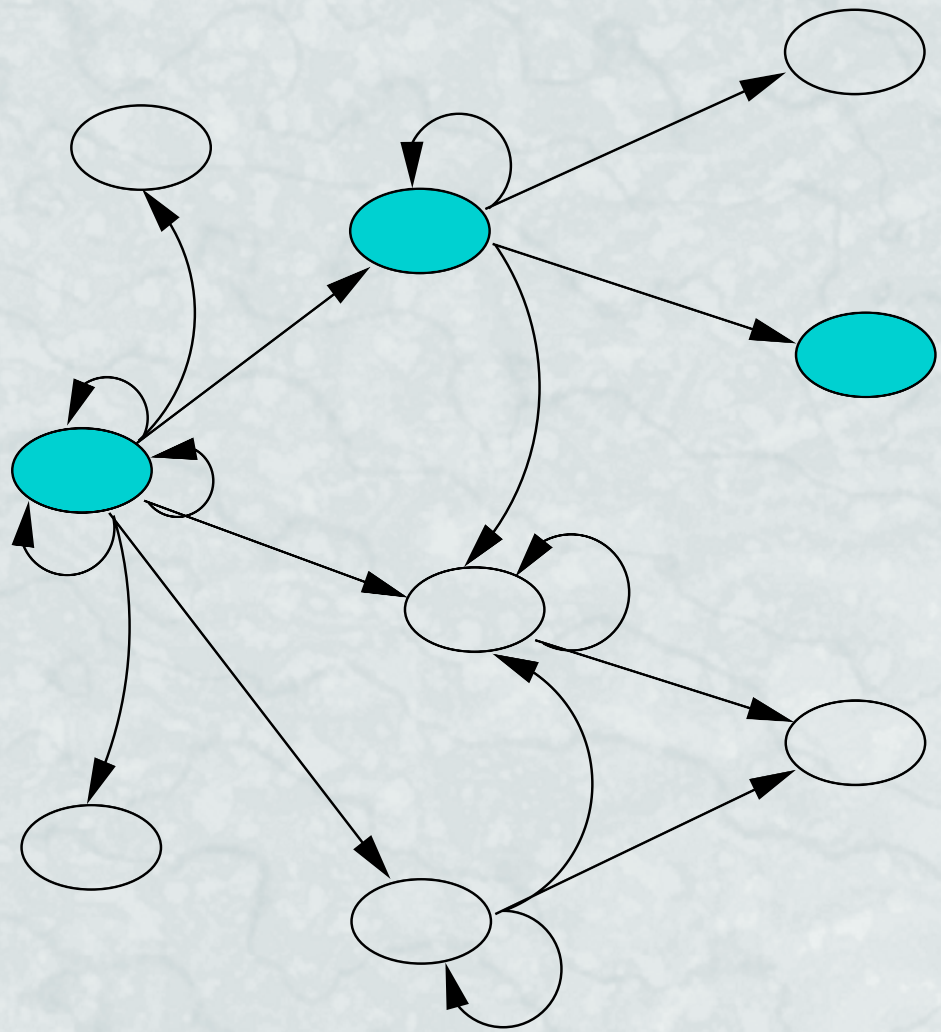
$$\Sigma = (S, A, P)$$

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:

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

- Expected value comes from a reward function
- An optimal policy is a policy that maximizes the expected value of every state

$$\pi^*(s) = \arg\max_{a \in A(s)} \left[u(a,s) + \sum_{s' \in S} \Pr_a(s' \mid s) V^*(s') \right]$$

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

state0 → actionB

state1 → actionD

state2 → actionA

States

Hierarchical Task Network

- Not enumerated exhaustively
- State consists of properties of the environment

$$vehicle(humvee, h1) \wedge vehicle(apc, a2)$$

- 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

State Space Size

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

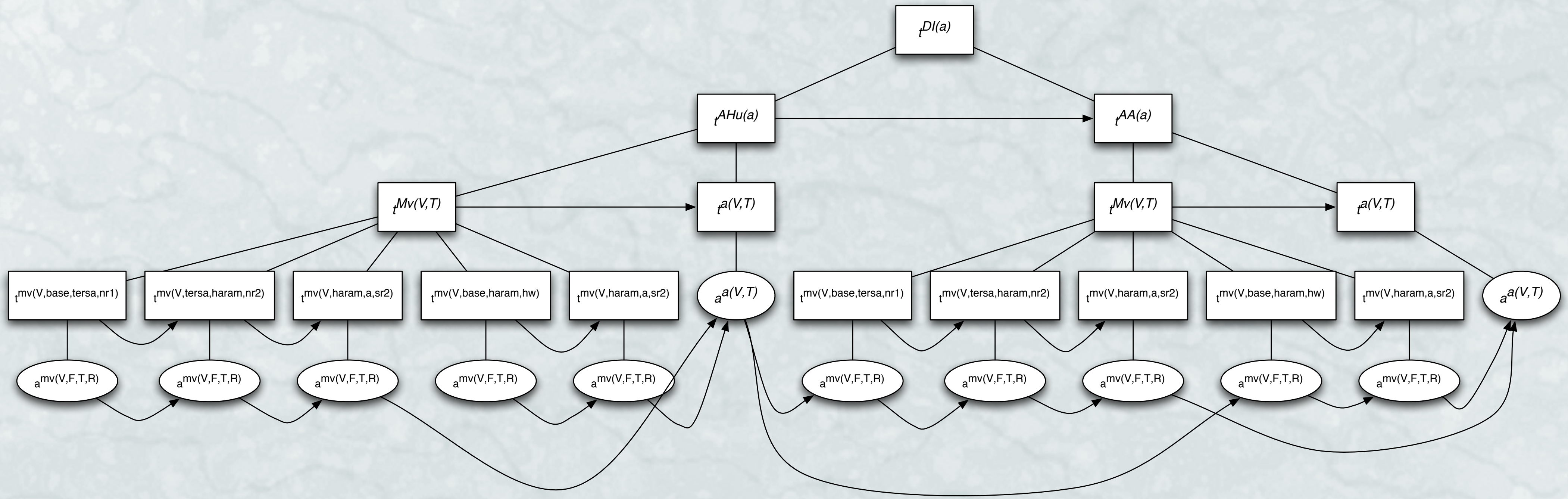
Markov Decision Process

- MDP solver must consult the *entire* state space
- State-space reduction techniques include:
  - Factorization
  - ϵ-homogeneous aggregation

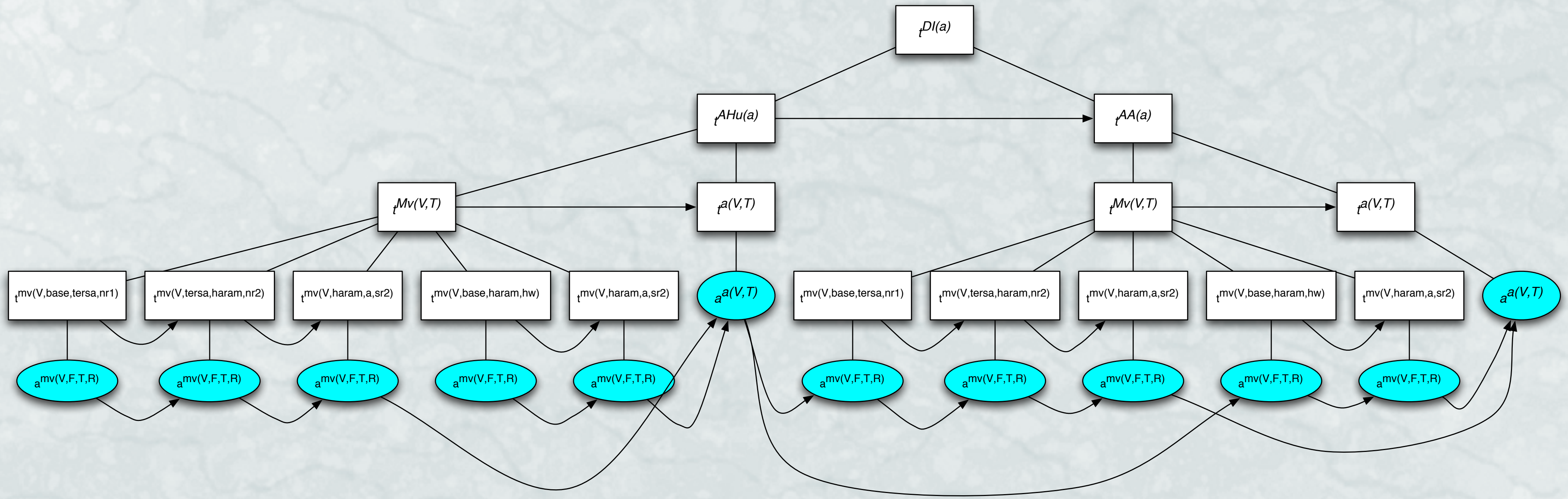
HTNs to represent MDPs

- 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

Fully Expanded HTN



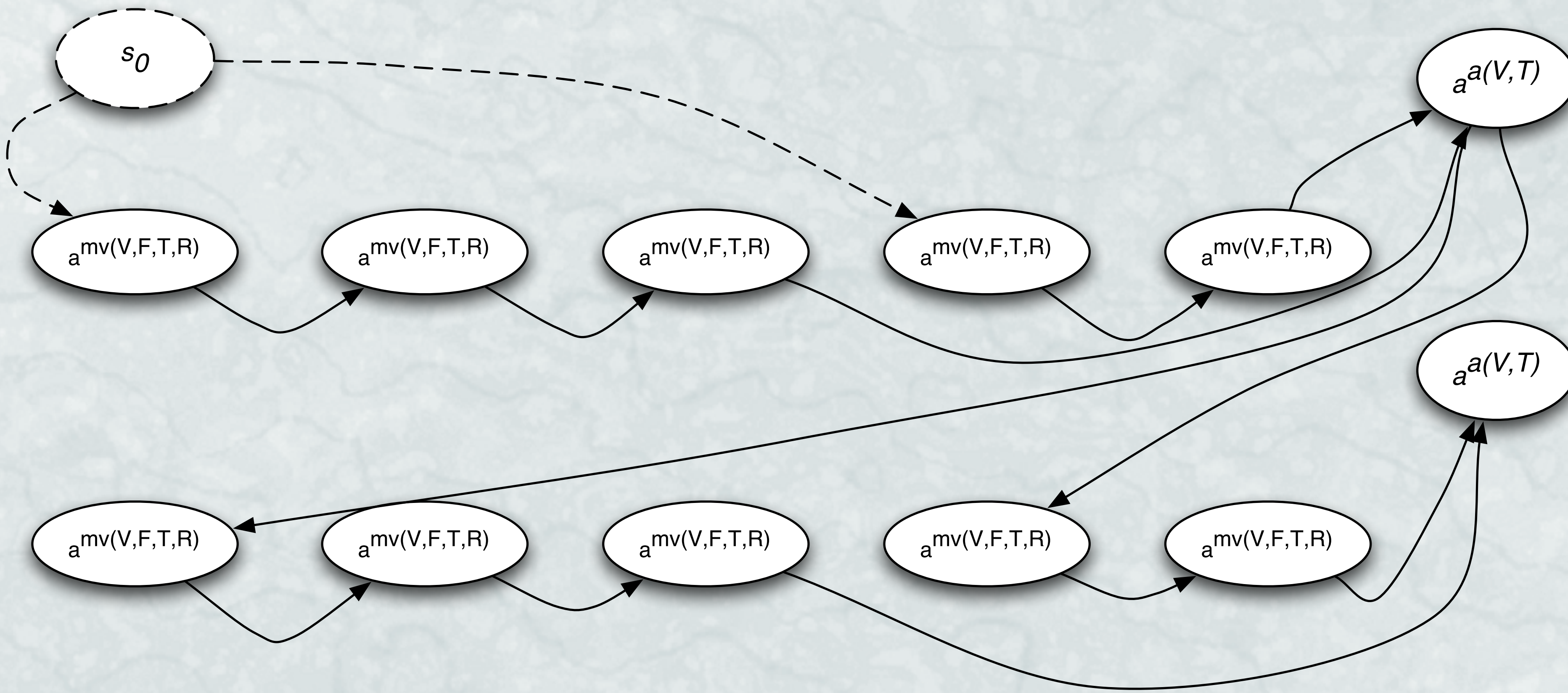
Reachable states



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



Conversion example

	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14
t0	0	0.5	0	0	0	0.5	0	0	0	0	0	0	0	0	0
t1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
t2	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
t3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t4	0	0	0	0	0	0	0	0	0.5	0	0	0	0.5	0	0
t5	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
t6	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
t7	0	0	0	0	0	0	0	0	0.5	0	0	0	0.5	0	0
t8	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
t9	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
t10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t12	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
t13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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 ϵ-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?