Minimizing the Information Leakage Regarding High-Level Task Specifications
Motivation

• By only completing a single task, the objective of an agent is clear to an observer.
• By completing multiple tasks, an observer must attempt to infer the true objective of the agent.
  • Uncertainty in which task the agent cares about completing, observer cannot optimally allocate resources
High-Level Problem Formulation

• The agent and the observer each know a set of specifications, $\Phi = \{\phi_1, \ldots, \phi_N\}$.
  - Only the agent knows the ground-truth specification, $\phi^* \in \Phi$.
  - The observer seeks to infer $\phi^*$ from the set of candidate specifications $\Phi_{can} \subseteq \Phi$, the specifications completed with a probability above some common-knowledge threshold.

• **Goal:** Synthesize the agent’s policy such that it completes $\phi^*$ with a desired probability while also leading an observer to believe that each $\phi_i \in \Phi_{can}$ is equally likely to be $\phi^*$.

$$\Phi = \{\text{deliver to blue}, \text{deliver to green}\}$$
Agent Model

- The agent operates in a stochastic environment modelled as a **Markov decision process (MDP)**, given by $\mathcal{M} = \{S, s_0, \mathcal{A}, \mathcal{P}, \mathcal{AP}, \mathcal{L}\}$
  - $S$ is a finite set of states
  - $s_0$ is a unique initial state
  - $\mathcal{A}$ is a finite set of actions
  - $\mathcal{P}$ is a transition function, $\mathcal{P}: S \times \mathcal{A} \rightarrow \Delta(S')$
  - $\mathcal{AP}$ is a set of atomic propositions
  - $\mathcal{L}$ is a labelling function, $\mathcal{L}: S \rightarrow 2^{\mathcal{AP}}$

- A **policy** for an agent is a sequence $\pi = (d_1, d_2, ...)$ where each $d_t: S \rightarrow \Delta(\mathcal{A})$. Denote the set of all policies by $\Pi(\mathcal{M})$.
  
  $\Delta(\cdot)$ is a probability mass function
Agent Specifications

• Use temporal logic to express agent specifications.
  • Relate occurrence of an event, causality between events, and ordering of successive events

• Focus on syntactically co-safe parametric linear temporal logic, includes parameterized temporal operators:
  • Parameterized Always: $\square_{[a,b]} p$
  • Parameterized Eventually: $\Diamond_{[a,b]} p$
  • Can nest together: $\square_{[a,b]} \Diamond_{[c,d]} p$

Eventually, between the third and fifth time steps, visit the blue base

\[ \Diamond_{[3,5]} \text{blue} \]
Observer Inference Model

- Ignores specifications satisfied with low probability when inferring which is $\phi^*$
- Uses a simple \textit{averaging rule} to assign inference probabilities:

$$\Pr(\phi = \phi^* | \phi \in \Phi_{can}) := \frac{\sum_{\phi \in \Phi} \Pr_{\pi_M}(s_0 \models \phi) \mathbb{I}\{\phi \in \Phi_{can}\}}{\sum_{\phi \in \Phi} \Pr_{\pi_M}(s_0 \models \phi_i) \mathbb{I}\{\phi \in \Phi_{can}\}}$$

  - Probability of satisfying $\phi$ in MDP $M$ under policy $\pi$
  - Indicator function taking value 1 if $\phi \in \Phi_{can}$ and 0 otherwise.
  - Probability of satisfying $\phi$, ignore if $\phi \notin \Phi_{can}$

- Uncertainty of the observer measured using the \textit{entropy} of its inference probabilities.

$$H^\pi(\Phi_{can}) := - \sum_{\phi \in \Phi_{can}} \Pr(\phi = \phi^* | \phi \in \Phi_{can}) \log(\Pr(\phi = \phi^* | \phi \in \Phi_{can}))$$
Problem: Synthesis of Minimum Information Leakage Policy

- Synthesize a policy $\pi \in \Pi(\mathcal{M})$ for the agent solving:

$$\max_{\pi \in \Pi(\mathcal{M})} H^\pi(\Phi_{can})$$

subject to: $Pr^\pi_\mathcal{M}(s_0 \models \phi^*) \geq \Gamma$

$$\phi \in \Phi_{can} \iff Pr^\pi_\mathcal{M}(s_0 \models \phi) \geq \beta$$

- $\mathcal{M}$: given MDP
- $\phi^*$: given ground-truth specification

Agent must satisfy $\phi^*$ with probability above this threshold for observer to consider it a candidate specification

Agent must satisfy $\phi_i$ with probability above this threshold for observer to consider it a candidate specification
Solution – Product MDP

- For each specification $\phi_i$, construct its corresponding deterministic finite automaton (DFA), $A_i$.
- To determine the satisfaction probabilities, form the product MDP $\mathcal{M} \otimes A_i$ for each $i = 1 \ldots N$. 

\[ a, 0.1 \quad a, 0.9 \quad a, 1 \quad b, 0.5 \quad b, 1 \quad b, 0.5 \]

\[ s_0 \quad s_1 \]

\[ s_0 \to s_1 \text{ with } a \text{ and } b \]

\[ s_1 \text{ is the unique initial state} \]

\[ A_i \]

\[ q_0 \to q_1 \to q_2 \to q_i \quad \text{with } \phi_i = \square_{\leq 2} \neg \text{blue} \]

\[ q_0 \to q_3 \text{ with } \text{blue} \]

\[ q_0 \text{ is the accepting state} \]

\[ \mathcal{M} \otimes A_i \]

\[ \times \quad \times \cdots \]
Solution – “Exact Approximation” of the Objective

\[ H^\pi(\Phi_{can}) := - \sum_{\phi \in \Phi_{can}} \Pr(\phi = \phi^* | \phi \in \Phi_{can}) \log(\Pr(\phi = \phi^* | \phi \in \Phi_{can})) \]

\[ = \frac{\nu(i)}{\sum \nu(i)} \text{ where } \nu(i) = Pr^\pi_M(s_0 \models \phi) \mathbb{I}\{\phi \in \Phi_{can}\} \]

- Replace this bilinear constraint by a **McCormick Envelope** (affine in \( Pr^\pi_M(s_0 \models \phi) \), \( \mathbb{I}\{\phi \in \Phi_{can}\} \)):

  - \( \nu(i) \leq \mathbb{I}\{\phi \in \Phi_{can}\} \)
  - \( \nu(i) \leq Pr^\pi_M(s_0 \models \phi) \)
  - \( \nu(i) \geq 0 \)
  - \( \nu(i) \geq \mathbb{I}\{\phi \in \Phi_{can}\} + Pr^\pi_M(s_0 \models \phi) - 1 \)

- The relaxation is exact, \( \nu(i) = Pr^\pi_M(s_0 \models \phi) \text{ if } \mathbb{I}\{\phi \in \Phi_{can}\} = 1 \) and 0 otherwise.
Solution – Optimization Problem

- Mixed-integer program (MIP) with quasi-concave objective function

\[
\max_{\lambda(s,a)x(i)} \left( \sum_{i \in [N]} \nu(i) - \sum_{j \in [N]} \nu(j) \log \left( \frac{\nu(i)}{\sum_{j \in [N]} \nu(j)} \right) \right)
\]

subject to:

\[
\forall s \in S_p \setminus B, \quad \lambda(s,a) - \sum_{s' \in S_p} \sum_{a' \in \mathcal{A}} \mathbb{P}_{s',a,s} \lambda(s',a) = \alpha(s)
\]

\[
\forall i \in [N], \quad \mu_i = \sum_{s \in S_p:s \in [i+2]} \sum_{a \in \mathcal{A}} \lambda(s,a)
\]

\[
\mu(1) \geq \Gamma
\]

\[
\forall i \in [N], \quad \mu(i) \geq \beta x(i)
\]

\[
\forall i \in [N], \quad v(i) = \mu(i)x(i)
\]

\[
\forall i \in [N], \quad x(i) \in \{0,1\}
\]

\[
\forall s \in S_p, \forall a \in \mathcal{A}, \quad \lambda(s,a) \geq 0
\]

\[
\pi(s,a) = \frac{\lambda(s,a)}{\sum_{a \in \mathcal{A}} \lambda(s,a)}
\]

- Solve using bisection method together with MIP solvers

- State space is the product of the MDP with all specification DFAs

Entropy of observer’s likelihood probabilities
Flow constraint: If you visit a state, must also leave that state
Probability of satisfying the i-th specification, \( \phi_i \)
Must satisfy \( \phi^* \) with probability \( \Gamma \)
(Assume \( \phi^* = \phi_1 \))
Only consider satisfaction probability if \( \phi_i \in \phi_{can} \). Use McCormick Envelope for middle constraint.
Expected residence is non-negative
Afterwards, obtain policy from flow variables
Approximate Solution Method: Probabilities at Each Time Step

- Predict whether a specification holds from whether its atomic propositions hold at each time step.
- Use the Fréchet inequalities to relate the probabilities of satisfying each $\phi_i$ at an individual time step to the probability of satisfying $\phi_i$ over the entire time interval.

$\phi_i = \Box_{[k_1,k_2]} p_i$: $p_i$ should hold at every time step over $[k_1, k_2]$.

Fréchet inequality for conjunction:
\[
\Pr\left(\bigwedge_{t=k_1}^{k_2} p_i(t)\right) \geq \max\{0, \sum_{t=k_1}^{k_2} \eta_i(t) - (k_2 - k_1)\}
\]

$Pr_{\mathcal{M}}^\pi(s_0 \models \phi = \Box_{[k_1,k_2]} p_i)$

$\phi_i = \Diamond_{[k_1,k_2]} p_i$: $p_i$ should hold at least once over $[k_1, k_2]$.

Fréchet inequality for disjunction:
\[
\Pr\left(\bigvee_{t=k_1}^{k_2} p_i(t)\right) \geq \max\{\eta(k_1), \ldots, \eta(k_2)\}
\]

$Pr_{\mathcal{M}}^\pi(s_0 \models \phi = \Diamond_{[k_1,k_2]} p_i)$
Approximate Solution – Optimization Problem

\[
\max_{\lambda(s,a,x)} \left( \frac{1}{2} \nu(i) \sum_{j \in [N]} \frac{\nu(i)}{\nu(j)} \log \left( \frac{\nu(i)}{\sum_{j \in [N]} \nu(j)} \right) \right)
\]

subject to:

\[
\forall s \in S_p \setminus B, \quad \sum_{a \in A} \lambda(s,a) - \sum_{s' \in S_p} \sum_{a \in A} \mathbb{P}_{s',a,s} \lambda(s',a) = \alpha(s)
\]

\[
\forall i \in [N], \quad \sum_{s \in S^{[t+1]}_{i}} \sum_{a \in A} \lambda(s,a) = \eta_i(t)
\]

\[
\forall i \in [N], \quad \mu_i = \sum_{s \in S_p : s^{[t+2]} \in F_i} \sum_{a \in A} \lambda(s,a)
\]

\[
\mu(1) \geq \Gamma
\]

\[
\forall i \in [N], \quad \mu(i) \geq \beta x(i)
\]

\[
\forall i \in [N], \quad \nu(i) = \mu(i) x(i)
\]

\[
\forall i \in [N], \quad x(i) \in \{0,1\}
\]

\[
\forall s \in S_p, \forall a \in A, \quad \lambda(s,a) \geq 0
\]

\[
\pi(s,a) = \frac{\lambda(s,a)}{\sum_{a \in A} \lambda(s,a)}
\]

- Remains a MIP with a quasi-concave objective function.

- The program uses the state space of an expanded MDP (but not the product of many automata).

For each specification, replace right-hand side with lower bounds derived from Fréchet inequalities.

For each specification, determine the probability that it holds at each time step over interval of interest.
Example – Delivering Supplies to Bases

- Agent must resupply a specific base at a specific time over the timespan of interest

- Consider two different sets of specifications – require the agent to visit different numbers of bases

<table>
<thead>
<tr>
<th>Example</th>
<th>Specifications $\phi$</th>
<th>Num. Vars. (Continuous, binary)</th>
<th>Sol. Time, Exact</th>
<th>Sol. Time, approx.</th>
<th>$H^\pi(\phi_{\text{can}})$, exact</th>
<th>$H^\pi(\phi_{\text{can}})$, approx</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resupply-1</td>
<td>$\phi^* = \mathbb{[9,10]}<em>{\text{blue}}$ $\phi_2 = \mathbb{[29,30]}</em>{\text{red}}$</td>
<td>Exact: 4870 con. 1 bin.</td>
<td>9.25 s</td>
<td>6.75 s</td>
<td>1.000 bits</td>
<td>0.999 bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Approx: 3079 con. 1 bin.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resupply-2</td>
<td>$\phi^* = \mathbb{[9,10]}<em>{\text{blue}}$ $\phi_2 = \mathbb{[16,18]}</em>{\text{Yellow}}$ $\phi_3 = \mathbb{[23,25]}<em>{\text{green}}$ $\phi_4 = \mathbb{[29,30]}</em>{\text{red}}$</td>
<td>Exact: 6980 con. 3 bin.</td>
<td>53.71 s</td>
<td>25.11 s</td>
<td>1.999 bits</td>
<td>1.999 bits</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Approx: 3125 con. 3 bin.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For both sets of specifications, each method maximizes the uncertainty of the adversary.

$\Gamma = 0.95$, $\beta = 0.80$. 
Example – Patrolling a Perimeter

- One outpost is assumed to store critical information. Based on patrolling of agent, observer seeks to infer the outpost using agent’s probability of visiting.

- Agent must repeatedly visit each outpost over time span, attempt to do so with equal probability for each.

<table>
<thead>
<tr>
<th>Example</th>
<th>Specifications $\phi$</th>
<th>Num. Vars. (Continuous, binary)</th>
<th>Sol. Time, Exact</th>
<th>Sol. Time, approx.</th>
<th>$H^n(\phi_{can})$, exact</th>
<th>$H^n(\phi_{can})$, approx</th>
</tr>
</thead>
<tbody>
<tr>
<td>Surveillance</td>
<td>$\phi^* = \Box_{[1,10]} \circ_{[0,5]} \text{upper left}$</td>
<td>Exact: 22209 con. 1 bin.</td>
<td>148.37 s</td>
<td>26.59 s</td>
<td>1.999 bits</td>
<td>1.999 bits</td>
</tr>
<tr>
<td></td>
<td>$\phi_1 = \Box_{[1,10]} \circ_{[0,5]} \text{upper right}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\phi_2 = \Box_{[1,10]} \circ_{[0,5]} \text{lower left}$</td>
<td>Approx: 617 con. 239 bin.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\phi_3 = \Box_{[1,10]} \circ_{[0,5]} \text{lower right}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Even with a large number of binary variables (needed to utilize off-the-shelf solvers), the approximate solution method is quicker and performs as well as the exact solution method.
Summary

- Considered the problem of minimizing the ability of an observer to predict the specification that an agent \textit{actually} cares about completing.
- Developed two algorithms, an exact and an approximate solution method to synthesize a policy for the agent.
  - Exact solution method gives exact satisfaction probabilities but can be cumbersome.
  - Approximate solution is quicker but may underreport the set of candidate specifications.

Eventually, between the third and fifth time steps, visit the blue base.