Topological Abstraction and Planning for the Pursuit-Evasion Problem

Alberto Speranzon*, Siddharth Srivastava (UTRC)
Robert Ghrist, Vidit Nanda (UPenn)
IMA 1/1/2015

*Now at Honeywell Aerospace – Advanced Technology

This material is based upon work supported by Defense Advanced Research Project Agency under contract No. HR0011-15-C-0043

Sponsored by the Defense Advanced Research Project Agency Defense Sciences Office (DSO)
Program: DARPA Seedling Program
Purchase request No.: HR0011513129, Program code: DMSSD Issued by DARPA/CMO under contract No. HR0011-15-C-0043

The views and conclusions contained in this presentation are those of the authors and should not be interpreted as representing the official policies, either expressly or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.
Consider a **complex planning** problem:
- Mission level problem – what the whole system should do
- Motion level problem – what the single sub-systems do
- Control level problem – generate kinematically/dynamically feasible motions

It is generally convenient to approach the design considering a **hierarchical control structure**

We consider **pursuit-evasion problem (P/E)** as a case study.
Complex Planning Problems: Hierarchical Control

- Consider a complex planning problem:
  - Mission level problem – what the whole system should do
  - Motion level problem – what the single sub-systems do
  - Control level problem – generate kinematically/dynamically feasible motions

- It is generally convenient to approach the design considering a hierarchical control structure

- We consider pursuit-evasion problem (P/E) as a case study
Assumptions of P/E Problem in this Talk

- **Multiple evaders** in the domain
- **Unknown number of evaders** in the domain
- **Different sensor capabilities** of the pursuers
  - Sensor footprint
  - Multiple pursuers to detect one evader
- **Limited range** of pursuers’ sensors
- **Communication constraints**: not all-to-all
- **Deterministic models**: position of the pursuers and sensing
- Known environment but **not restricted to 2D**, with obstacles
Contents

- Sequential decision making
- Pursuit/Evasion and topological abstractions
- Decision making over sheaves
- Generalization of the P/E with sheaves
- Nerve complex and downward refinement
- Conclusions
Sequential Decision Making

- We consider the special case: deterministic, fully observable, i.e. an MDP

- **Given transition system** $\mathcal{T} = (S, A, T, S_{goal}, S_{init})$:
  - $S$: Set of states,
  - $A$: set of action symbols
  - $T$: $S \times A \rightarrow S$, deterministic transition function
  - Desired set of states $S_{goal}$

- **Compute**: “strategy” for reaching a desired state

- Forms of solutions:
  - Finite state machine
  - Policy: State $\rightarrow$ Action
Different level of abstractions have been considered in the problem of P/E:

- **Continuous versions**: generally posed as a differential game requiring the solution of Hamilton-Jacobi-Bellman-Isaac equation.
  - Solutions available only for simple models of P’s and E’s and environments

- **Discrete versions**: generation of a discrete representation, typically as a graph – grid or visibility graph. Planning is then computed as path along such graph.

**Several assumptions:**
- Configuration space is the same as the map space
- Sensor footprint matches the graph/grid resolution
- Downward “refineability” holds

Result in post-facto engineering for fixing the problem with on-the-fly replanning
We consider a **topological abstraction** of the problem.

**Ad-hoc** abstraction using a grid

Topological Abstraction

Pursuer space $\mathcal{P}$

Evader space $\mathcal{E}$

- Pursuer space $\mathcal{P}$: 2 connected components with 3 holes
- Evader Space $\mathcal{E}$: 4 connected components
P/E New Perspective on Abstraction

- We consider a **topological abstraction** of the problem
  
  - The topological abstraction is "minimal" as it encodes the critical information about the problem – 4 connected areas where evaders can be hiding. However, **how can we make it useful to generate actionable plans?**
  
- We need a way to **encode more (metric) information** about the overall problem

**Sheaf** as a way to capture information necessary for planning
Sheaf, Restriction Maps and Global Sections

- A sheaf is a mathematical object that stores **locally-defined data** over a space and allows to infer about **global properties** of the space from such data.

**EXAMPLE:**

![Space: \( \mathcal{X} \)]
A sheaf is a mathematical object that stores **locally-defined data** over a space and allows to infer about **global properties** of the space from such data.

**EXAMPLE:**

![Diagram of a space and complex](image)
A sheaf is a mathematical object that stores **locally-defined data** over a space and allows to infer about **global properties** of the space from such data.

**EXAMPLE:**

**Restrictions maps**, are maps between two cells \( \sigma \) and \( \tau \) where \( \sigma \) is a face of \( \tau \)

**EXAMPLE OF RESTRICTIONS MAPS:**

\[ A_{12}^{1} : \text{linear map between } \sigma = \{1\} \text{ and } \tau = \{12\} \]

\[ A_{12}^{2} : \text{linear map between } \sigma = \{2\} \text{ and } \tau = \{12\} \]
Simple (Familiar) Example

- Consider discrete points (0-simplexes) at certain time instances
- Assign vector spaces to each 0-simplex
- And to each 1-simplex as well

In this particular example, the sheaf co-homology represents the evolution of a linear time varying (or invariant if all the restriction maps are constant) system:

\[ u_{n+1} = A_n u_n \]
There are a number of compelling tools suited to reason about abstractions based on sheaves:

**GLOBAL SECTION FUNCTOR, \( H^0(-) \)**

Collates all solutions to the constraints imposed by the sheaf stalks and restriction maps; gives an algebraic form to the solution set.

In P/E, \( H^0(-) \) classifies connectivity of evasion paths.

**COHOMOLOGY, \( H^\bullet(-) \)**

\( H^\bullet(-) \) characterizes constraint satisfaction as a function of both the domain topology and the algebra of the constraints.

In P/E, this gives necessary and/or sufficient criteria for evasion.

**PUSHFORWARDS/PULLBACKS**

These operations transform sheaves from one base space to another.

In P/E, these allow for dynamic updates to domain/parameters.

**HOM AND TENSOR PRODUCTS**

HOM classifies relationships between sheaves; \( \otimes \) convolves sheaf data.

In P/E, HOM gives a collation of constraints; \( \otimes \) acts as sensor fusion.
Sheaf Abstraction of P/E Problems: Evader Sheaf
Sheaf Abstraction of P/E Problems: Evader Sheaf

Action: (up, right)

(Merging)
Sheaf Abstraction of P/E Problems: Evader Sheaf
Sheaf Abstraction of P/E Problems: Evader Sheaf

Action: (up, right)

Action: (up, right)

Action: (down, up)
Sheaf Abstraction of P/E Problems: Evader Sheaf

Compress evasion region to connected components

- Use $H^0(\_; \mathbb{R})$ to have sheaf take values in real vector spaces
- Use (persistent) co/homology to track evader components
- Note that at this level of abstraction we are considering an abstract state where each state encodes the number of connected components of $\mathcal{E}$
- Here we depict one sheaf for one set of actions, but we have multiple sheaves for each possible set of actions
In this simple setting, the sheaf captures the evolution of the connected components of the evader space.

Restriction maps in this context (can be made) are maps between vector spaces.
For each path in this search space we can associate a sheaf corresponding to a sequence of actions $\pi = \{A_{i_1}, A_{i_2}, A_{i_3}, \ldots, A_{i_n}\}$.

Evader capture in this context corresponds to the fact that $\dim H^0(-; \mathbb{R}) = 0$ or equivalently that the (evader) sheaf has no global section.
Simpler Example: Global Section in P/E

- Note that in this case there is no continuous path from “left” to “right”
- This indicates that there are no global sections in the sheaf...
- ... and equivalently that the sequence of actions corresponds to a capture
- Thus we have that in the P/E problem, the evader is captured if there exists a set of actions such that the corresponding sheaf has no global sections
Simpler Example: Global Section in P/E

- Note that in this case there is no continuous path from “left” to “right”
- This indicates that there are no global sections in the sheaf ...
- ... and equivalently that the sequence of actions corresponds to a capture
- Thus we have that in the P/E problem, the evader is captured if there exists a set of actions such that the corresponding sheaf has no global sections

Sheaves provide a very general framework to approach this problem as we can encode arbitrary data structures on the sheaves thus capturing richer settings
So What Have We Done?

**Agent Model Abstraction**

- Sensing
- Agent Motion
- Abstract Model

**Agent Model Abstraction Over a Domain**

**Nerve Construction**

**Evader Sheaf and Planning**

**Reduce Transition System Through State Abstraction**

**Transition System**
Extensions Using Sheaf Theoretic Methods

Domain Size & Topology

Evader Capture Criteria

Communication Network

Temporal Dynamics

Sensing & evasion sheaves

Augment sheaf stalks; build \textit{escape and capture complex}

Factor into base space as \textit{communication complex}

Sheaves of semigroups; duality

\begin{align*}
E_1 &= P_1 P_2 + P_1 P_3 + P_4 \\
E_2 &= P_2 P_3 + P_2 P_4 \\
E_3 &= P_1 P_2 P_3 + P_1 P_3 P_4
\end{align*}
Capture Constraints: Propositional Logic

Each evader is assigned its own capture criterion in the form of a **Boolean expression** involving (some of) the pursuers:

\[
E_1 = P_1 P_2 + P_1 P_3 + P_4 \\
E_2 = P_2 P_3 + P_2 P_4 \\
E_3 = P_1 P_2 P_3 + P_1 P_3 P_4
\]

In this example, the first evader can be caught only if one of the three following conditions it satisfied – it must be:

- seen simultaneously by pursuers $P_1$ and $P_2$, or
- seen simultaneously by pursuers $P_1$ and $P_3$, or
- seen by pursuer $P_4$ alone

On the other hand, the third evader needs to be simultaneously seen by pursuers $P_1$ and $P_3$ along with one of $P_2$ or $P_4$.

In this manner, we can encode vastly asymmetric models of pursuer and evader capabilities.

**MULTIPLICATION $\leftrightarrow$ AND : ADDITION $\leftrightarrow$ OR**
Evasion and Capture Sheaf

Idea: Encode Boolean capture criteria among pursuers/evaders as a complex. Use this complex as the base of an *Evasion Sheaf*, to enforce AND operations. Define a secondary *Capture Sheaf*, to enforce OR operations.

Given: Sensed (coverage) regions $C_{i,t}$ in $\mathbb{R}^n$ as a function of time $t$ (action sequence) and pursuer $P_i$ positions; Boolean statements for capture criteria among pursuers/evaders.

Base space: product of action sequence ($\mathbb{R}$) with *Escape Complex*

**DISCUSSION FOLLOWS THE SIMPLEST TIME-INDEPENDENT CASE**

NOVEL CONSTRUCTION

The *Escape Complex* is the subcomplex of the full simplex on the pursuer set $\{P_i\}$ defined as the maximal subcomplex of the simplex not intersecting the (open) cells defined by the Boolean pursuer relations.
The **Escape Complex** is the subcomplex of the full simplex on the pursuer set \{P_i\} that consists of the largest subcomplex of the simplex not intersecting the (open) cells defined by the Boolean pursuer relations.

**EXAMPLE:** If the capture criterion is...

\[ E = P_1P_2 + P_1P_3 = P_1(P_2 + P_3) \]

(capture requires participation of \(P_1\) and either \(P_2\) or \(P_3\))
Capture Criteria: AND

EXAMPLE: If the capture criterion is... \( E = P_1P_2 + P_1P_3 = P_1(P_2 + P_3) \)

Let the domain be a simple interval with four nodes (evader regions)

At a fixed time let the pursuer coverage regions be:

Build the evasion sheaf as follows:

*Stalks* = vector space whose dimension over vertices records size of uncovered region

*Sheaf maps* = inclusions identifying common regions
Cohomology over the escape complex tracks the **AND** operations in escape regions

In this simple example, one easily computes that \( \dim H^0 = 2 \).

One global section comes from what lies in the complement of \( P_1 \)'s coverage... 

The other global section is supported over the \( P_2 - P_3 \) edge and is generated by the intersection of complements of \( P_2 \) and \( P_3 \) coverage regions.

This yields the **AND** operations in the factored Boolean expression

\[
E = P_1 P_2 + P_1 P_3 = P_1 (P_2 + P_3)
\]
Capture Criteria: OR

Express the **OR** as a cohomology over another complex, the **Capture Complex**

In this simple example, we need to intersect capture by $P_1$ with capture by $(P_2 \text{ OR } P_3)$

Consider the simplicial complex generated by these two conditions in the AND.

Repeat the construction: take complements of the regions implicit in $H^0$ of the evasion sheaf; use these as stalks of the Boolean capture sheaf, then compute $H^0$.

In this example, $\dim H^0(CAP; H^0(ESC)) = 2$ – the two locations of capture.

In general, this yields the **OR** operations in the Boolean expression

$$E = P_1P_2 + P_1P_3 = P_1(P_2 + P_3)$$
Communication Constraints

We can accommodate complex constraints on how the pursuers are permitted to communicate with one another.

These constraints are represented by a simplicial complex whose vertices index the pursuers. An edge implies communication that is pairwise – higher collaboration comes from higher simplices.

All planning and coordination become subordinate to the structure of this communication complex: in the maximal case, one has full communication and a full simplex; in the worst case, there are only vertices and no communication.

We assume a model where communication leads to symmetric sharing of information about coverage; this corresponds to taking an OR operation with coverage.
**Communication Complex**

**Idea:** Encode communication constraints among pursuers as a simplicial complex

**Given:** Sensed (coverage) region $C_i$ in $\mathbb{R}^n$ as a function of pursuer $P_i$; Communication constraints as complex on pursuer set.

**Base space:** product of time axis (action sequence) $\mathbb{R}$ with **Communication Complex**

**Stalks:** begin with $\mathbb{R}^n - C_i$, i.e. the non-sensed regions where evaders can hide; convert these to vector spaces whose dimensions over vertices give size of uncovered region

- $P_3$ and $P_4$ can communicate
- $P_1$, $P_2$ and $P_3$ can communicate with each other
EXAMPLE: Consider a collection of three pursuers viewing a circular domain.

In this example, the cohomology $H^0 = 0$, since the lower $P_2$-$P_3$ edge has trivial kernel. In other words, when everyone communicates, the evasion region vanishes – no escape.
EXAMPLE: Consider a collection of three pursuers viewing a circular domain.

In this example, the cohomology $H^0 = \mathbb{R}^4$ when $P_2$ cannot communicate, meaning that an evader can hide from $P_2$ (in three locations) or from $P_1$ and $P_3$ (in one).
Temporal Constraints

All of our data are allowed to vary with time. The directionality of time makes it difficult to classify evasion paths as topological constructs.

**sensed region, time-varying**

- **true** evasion path (global section of evader sheaf)
- **false** evasion path (violates causality)
Consider a complex planning problem:

- Mission level problem – what the whole system should do
- Motion level problem – what the single sub-systems do
- Control level problem – generate kinematically/dynamically feasible motions

It is generally convenient to approach the design considering a **hierarchical control structure**

We consider **pursuit-evasion problem (P/E)** as a case study.
Nerve Complex

- Consider a collection \( U = \{ U_\alpha \} \) of (compact) subsets of a topological space

- The nerve \( N(U) \) of \( U \) is built as follows:
  - The \( k \)-simplices of \( N(U) \) correspond to nonempty intersection of \( k + 1 \) distinct element of \( N(U) \)

EXAMPLE:
Agent Abstract Model

- Given:
  - Sensing model of agents
  - Motion model of agents
- We can build low-level model that capture various details:
Action Definition for 2D Environments

- Time is discretized
- Actions are determined from a minimal cover of the 1-step sensing envelop

\[ \mathcal{A} = \{a_1, a_2, \ldots, a_M\} \]
State Abstraction via Nerve

- Generate the transitions system $T$ from action $\mathcal{A}$ and state $\mathcal{S}$ for a given environment.

- Nerve complex is used to determine the number of connected components corresponding to various configurations.
State Abstraction via Nerve

- Generate the transition system $T$ from action $\mathcal{A}$ and state $\mathcal{S}$ for a specific environment

- For example: in the figure any of the three positions of the green pursuer or the blue pursuer will lead to the same number of connected components
State Abstraction via Nerve

- Generate the transition system $T$ from action $\mathcal{A}$ and state $\mathcal{S}$ for a specific environment.

- A sub-tree/sub-branch of the full $T$ can be ‘collapsed’ into a single state and action based on the number of connected components obtained using the nerve construction: $A'_1 = \{a_{i_1}, a_{i_2}, a_{i_3}, \ldots\}$
Advantages of Nerve Construction

Observations:

- Transition system $T$ needs to be constructed only once
- Transition system $T$ can be very large however we never use $T$ to compute a plan
- Transition system $T'$ is much smaller and suitable for planning
- Nerve construction generalizes the grid construction that is thus encompassed with this model
- Downward refinability easier to achieve within this more general model
Summary

Agent Model Abstraction

Agent Model Abstraction over a Domain

Nerve Construction

Evader Sheaf and Planning

Reduce Transition System through State Abstraction

Transition System
Conclusions

 “Semantic abstraction” of the pursuit-evasion problem using connected components of the evader space
 Connected states to actions through using a sheaf
 Topological properties (global sections) of the sheaf lead to evader capture conditions
 Nerve based construction to abstract low-level control laws

Left To Do (Everything! 😊)

 Global section is a too coarse notion: how can we select branches that are most “promising”?  
 How do we capture uncertainty?
 Computational aspects
 ....