Optimal Control of Mobile Sensor Networks

Silvia Ferrari
Laboratory of Intelligent Systems and Controls (LISC)
Department of Mechanical Engineering and Materials Science
Duke University

MAE Seminar Series
Department of Mechanical and Aerospace Engineering
Princeton University

November 20, 2009
Outline

- Introduction
- Motivation and Applications
- Optimal Control Framework
- Geometric Models of Mobile Sensors
- Cooperative Track Detection Problem
- Treasure Hunt Problem
- Marco Polo Problem
- Conclusions and Q&A
Modern Surveillance Systems – multiple sensors installed on mobile platforms
-- landmine detection and identification
-- monitoring of endangered species
-- monitoring of urban environments

Traditional paradigm: sensor information is used as feedback to the vehicle in order to support the vehicle navigation

New paradigm: the sensor motion is planned in view of the expected measurement process, in order to support the sensing objectives

LISC Research Emphasis: Geometric sensor path planning
-- Address couplings between sensor measurements and sensor dynamics
-- Optimize sensing objectives (e.g., detection, classification, tracking..)
Sensors: acoustic, w./ GPS, limited micro-level processing, mobile

Targets: passive, mobile, unauthorized

Environment: heterogeneous bathymetry and ambient properties, currents

Sensing objectives: coverage, tracking, detection, classification
Applications: UAV Demining

- Sensors: Cameras, IR, GPR, EMI, synthetic aperture radar (SAR)
- Targets: static, hidden, hazardous
- Environment: heterogeneous soils, weather, time of day, obstacles
- Sensing objectives: detection, classification
Applications: Urban Monitoring

- Sensors: Cameras, IR, synthetic aperture radar (SAR)
- Targets: static, hidden, mobile, evading
- Environment: time of day, obstacles
- Sensing objectives: detection, classification, tracking
Optimal Control Framework

**Performance measure** to be optimized w.r.t. $u(t)$,

$$J = \phi[\mathbf{x}(t_f), t_f] + \int_{t_0}^{t_f} L[\mathbf{x}(t), u(t), t]dt,$$  

with I.C. $\mathbf{x}(t_0)$

subject to nonlinear time-varying dynamics,

$$\dot{\mathbf{x}}(t) = f[\mathbf{x}(t), \mathbf{p}(t), u(t), t]$$

and subject to equality and inequality constraints

$$c[\mathbf{x}(t), u(t)] \geq 0$$

• Vehicles control vector: $u[\mathbf{x}(t), t]$
• Vehicles positions: $\mathbf{x}(t)$
• Environmental and sensing parameters: $\mathbf{p}(t)$
Modeling of Mobile Sensor Networks
Mobile Sensor Model

- The sensor is characterized by a discrete field-of-view (FOV) geometry, and by a joint probability density or mass function (PDF or PMF).
- The platform is characterized by a discrete vehicle geometry and a dynamic equation.

Examples:
Targets and Workspace

- The workspace may contain obstacles and changing environmental conditions
- Stationary targets are characterized by discrete geometries, and a prior probability density or mass function or prior: \( p(X^k), p(\lambda^k) \)
- Moving targets are characterized by a Markov motion model: \( p(x_j, v_j, \theta_j), \ j = 1, \ldots \)

Examples:
Probabilistic Model of Sensor Measurements

Classical Sensor Model (Estimation Theory):

\[ Z^k = h(X^k, \lambda^k) \]  

(Discrete time)

- Measurement vector: \( Z^k = [z_1(t_k) \ldots z_r(t_k)]^T \)  
- Target state: \( X^k = [x_1(t_k) \ldots x_n(t_k)]^T \)
- \((p \times 1)\)-Vector of sensor characteristics, e.g., noise and measurement errors: \( \lambda^k \)
- Deterministic, nonlinear vector function: \( h: \mathbb{R}^{n \times p} \rightarrow \mathbb{R}^r \)

Probabilistic Sensor Model:

\[ p(Z^k, X^k, \lambda^k) = p(Z^k \mid X^k, \lambda^k)p(X^k)p(\lambda^k) \]  

(Discrete time)

- Set of random measurement or observations at \( t_k \): \( Z^k \)  
- Set of random target state variables at \( t_k \): \( X^k \)
- Set of random sensor characteristics at \( t_k \): \( \lambda^k \)
- Joint probability density or mass function (PDF or PMF): \( p(Z^k, X^k, \lambda^k) \)
Underwater Sensor Networks for Cooperative Track Detection
Workspace and Environmental Conditions

Current Vector Field Over a 5-Day Period

Environmental conditions influence, $r$ (Km)

Real CODAR-Measured Current Field
(100 naut-NJ Coast)$^\dagger$

$^\dagger$[COOL, Rutgers University]
Underwater Vehicle Dynamics:

\[ M_i \ddot{\nu}_i + C_i(\nu_i) \dot{\nu}_i + D_i(\nu_i) \nu_i + g_i(\xi_i) = \tilde{B}_i \cdot T_i(\nu_i, u_i), \quad i = 1, \ldots, n \]

Vehicle's control bounds: \[ \| u \| \leq V_{\text{max}} \]

Sensor's effective range: \[ r_{\text{max}} = r(x, u, \alpha) \]

Simulation: sensor motion and effective range

Objective: Cooperative Track Detection

Track Coverage: Sensors’ ability to cooperatively detect target tracks

Track-before-detect Approach

Geometric Transversal Theory

- “Divide and conquer” algorithm for constructing the space of line transversals to $n$ line segments in $\mathbb{R}^2$ [Edelsbrunner, 1982].
- $O(r)$ algorithm for finding the slope of a line transversal to a family of $n$ disjoint convex translates [Egyed and Wenger, 1989].
- Finding a line transversal for a family of $n$ line segments or a family of $n$ circles in $\mathbb{R}^2$ takes $\Omega(n \log n)$ time on an algebraic decision tree [Avis et al., 1984].

Track Coverage Cones

- Represent space of line transversals for $k$ circles belonging to a family of $n$ (non-translates) circles as a function of their location in $A \in \mathbb{R}^2$.

**Approach:** Coverage cone ($k = 2$)

\[
\hat{h}_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix} \frac{v_i}{\|v_i\|} = Q_i^+ \hat{v}_i, \quad \hat{l}_i = Q_i^- \hat{v}_i.
\]
Redundant Track Coverage ($n > k$)

**Principle of union-exclusion** → union of possible non-disjoint sets

\[
\left| \bigcup_{i=1}^{n} A_i \right| = \sum_{j=0}^{n} \left( (-1)^{(j+1)} \sum_{S \in I_j} \right) \quad I_k = \text{set of } k \text{ fold intersections of members in } \{A_1, \ldots, A_n\}
\]

Example: $n = 3, k = 2$
• Opening angles: Lebesgue measure on the sets of line transversals.

Theorem: Probability of Track Detection

Theorem 3.6. The probability of detection of unobserved tracks for a set $\mathcal{P}$ of $N$ pursuers with fields-of-view $D_1, \ldots, D_N$, in a square game area $S$ of dimensions $L \times L$, is a multivariate probability density function of the sensors’ positions $X = \{p_1, \ldots, p_N\}$ given by a Lebesgue measure on this union,

$$
P_S^k(X) = \frac{\delta b}{4\pi L} \sum_{\ell=1}^{L/\delta b} \sum_{j=1}^{m} (-1)^{j+1} \sum_{1 \leq i_1 < \cdots < i_j \leq m} \left[ \psi \left( D_p^{i_1,j}, b_y^\ell \right) + \varphi \left( D_p^{i_1,j}, b_y^\ell \right) \right] \\
+ \frac{\delta b}{4\pi L} \sum_{\ell=0}^{(L/\delta b-1)} \sum_{j=1}^{m} (-1)^{j+1} \sum_{1 \leq i_1 < \cdots < i_j \leq m} \left[ \xi \left( D_p^{i_1,j}, b_x^\ell \right) + \rho \left( D_p^{i_1,j}, b_x^\ell \right) \right] \\
with \quad m = \frac{N!}{(N - k)!k!}, \quad D_p^{i_1,j} \equiv \left\{ D_k^{i_1} \cup \cdots \cup D_k^{i_j} \right\},
$$

where the summation $\sum_{1 \leq i_1 < \cdots < i_j \leq m}$ is a sum over all the $[m!/(m-j)!j!]$ distinct integer $j$-tuples $(i_1, \ldots, i_j)$ satisfying $1 \leq i_1 < \cdots < i_j \leq m$, $D_k^{i_i}$ denotes the $i_i$th $k$-subset of $D$, and $D_p^{i_1,j}$ is a $p$-subset of $D$, with $k \leq p \leq n$. 

Optimal Control Problem

**Sensing performance metric** to be optimized w.r.t. \( c[\cdot], x(\cdot) \)

\[
J = \phi[x(t_f), t_f] + \int_{t_0}^{t_f} \left\{ w_T \cdot P^k_S[x(t), r(t), u(t)] + w_E \cdot u^T(t)Ru(t) \right\} \, dt,
\]

subject to sensor network dynamics,

\[
x(t) = f[x(t), p(t), u(t), t]
\]

and subject to equality and inequality mission constraints

\[
g[x(t), u(t)] \geq 0
\]

- \( n \) vehicles controls: \( u(t) = c[x(t), t] \)
- \( n \) vehicles positions: \( x(t) \)
- Environmental and sensing parameters: \( p(t) \)
Optimal Sensor Network Trajectories

Comparison of Optimal Sensor Deployments

- $L_1 \times L_2 = 90 \times 82.5$ Km
- $\Delta T = 9$ days
- $n = 15$, $k = 3$,
- $r = 4$ Km
- ▼ UUV ● Buoys
Sensing Performance Comparison

**Track Coverage Time History**

Legend:
- : Optimal control and initial conditions ($x_0^*$)
- : Optimal control, given $x_0$
- : Optimal initial conditions ($x_0^*$), no control (buoys)
- : Static optimal
Extension to Maneuvering Targets
Track Coverage for Maneuvering Targets

Markov Target Model:

\[ \mathbf{x}(t) = \mathbf{x}_j + \mathbf{v}_j (t - t_j) \begin{bmatrix} \cos \theta_j \\ \sin \theta_j \end{bmatrix}^T, \quad \mathbf{x}_j = \mathbf{x}(t_j), \quad t_j \leq t < t_{j+1}, \quad j = 1, 2, \ldots \]

Markov motion parameter values: \( \{ \mathbf{x}_j, \mathbf{v}_j, \theta_j \}_{j=1,2,\ldots} \)

Both \( \mathbf{x} \) and \( \mathbf{s}_i \) are functions of time

Markov trajectory amounts to straight-line segments in the space $\mathbb{R}^2 \times [t_0, t_f]$
Heading and Velocity Cones

The 3D coverage cone $K[D_i, x_j]$ can be represented by its projection onto the $xy$-plane, $K_\theta$, and by $K_v$, which denotes its intersection with the velocity plane: $(\sin \theta_j)x + (\cos \theta_j)y = 0$

$K_v$ is finitely generated by the unit vectors,

$$\hat{\xi}_i(t) = \begin{bmatrix} \cos \theta_j \sin[\pi/2 - \eta_i(t)] \\ \sin \theta_j \sin[\pi/2 - \eta_i(t)] \\ \cos[\pi/2 - \eta_i(t)] \end{bmatrix} \quad \text{and} \quad \hat{\omega}_i(t) = \begin{bmatrix} \cos \theta_j \sin[\pi/2 - \mu_i(t)] \\ \sin \theta_j \sin[\pi/2 - \mu_i(t)] \\ \cos[\pi/2 - \mu_i(t)] \end{bmatrix}$$

where:

$$\eta_i, \mu_i = \tan^{-1}\left\{ t \left[ x_i \cos \theta_j + y_i \sin \theta_j + \sqrt{r_i^2 - (x_i \sin \theta_j + y_i \cos \theta_j)^2} \right]^{-1} \right\}$$

$K_\theta$ and $K_v$ are a function of the sensor coordinates $x_i(t)$ and $y_i(t)$ and of the Markov motion parameter values: $\{x_j, v_j, \theta_j\}_{j=1,2,\ldots}$
At every time $t_j \leq t < t_{j+1}$, the 3D spatio-temporal cone $K[D_i, x_j]$ contains all Markov pwl. tracks that originate at $x_j$ and are detected by the $i^{th}$ sensor at $x_i$. 
Example: Application of ST cone at $t = 60$ s

For a given layout $W \subseteq \mathbb{R}^2$ with $r$ targets and $n$ obstacles and a given joint probability mass function $P(y, m_1, \ldots, m_r)$ of an hypothesis variable, $y$, and $r$ measurements, find the obstacle-free path that minimizes the distance traveled by a robot $A$, between two configurations $q_0$ and $q_f$ and maximizes the information value for a sensor with field of view $S$, installed on $A$.

Definitions

Information Value: Expected Entropy Reduction (EER):

\[ \Delta H(X^k; Z^k | Z^1, ..., Z^{k-1}, \lambda^k) = H(X^k | Z^1, ..., Z^{k-1}, \lambda^k) - \sum_{Z^k} H(X^k | Z^1, ..., Z^k, \lambda^k) p(Z^k | Z^1, ..., Z^{k-1}, \lambda^k) \]

Definition 4.1 (Field of View): The field of view of a sensor mounted on \( A \) is a closed and bounded subset \( S(q) \subset \mathcal{W} \) such that the measurement set of a target located at any point \( p \in S(q) \) can be obtained by the sensor when the robot occupies the configuration \( q \in C \).

Definition 4.2 (C-Target): The target \( T_i \) in \( \mathcal{W} \) maps in the robot’s configuration space, \( C \), to the C-target region \( CT_i = \{ q \in C \mid S(q) \cap T_i \neq \emptyset \} \).

Definition 4.3: A void cell is a convex polygon \( \kappa \) in \( C_{\text{free}} \) with the property that none of the targets are observable from any of the configurations in \( \kappa \).

Definition 4.4: An observation cell is a convex polygon \( \bar{\kappa} \) in \( C_{\text{free}} \) with the property that every configuration in \( \bar{\kappa} \) enables a non-empty set of measurements \( Z(\bar{\kappa}) = \{ M_i \mid q \in \bar{\kappa}, q \in CT_i \} \).
Sensor Path Planning

- Develop a cell decomposition method that accounts for the geometries of the targets and the sensor FOV.

Connectivity graph $G$ with observation cells labeled in grey and void cells labeled in white.
Approximate-and-Decompose Method

- Configuration $q$ of robot $A(q) : q = (x, y, \theta)$ with orientation $\theta$
- Configuration space, $C$ : the space of all the possible configurations of $A$

Examples of C-obstacle, $CB$, and C-target, $CT$, obtained for a sensor with FOV, $S(q)$:

- Bounding rectangloid approximation $RB$ of $CB$
- Bounded rectangloid approximation $R'T$ of $CT$
1. Decompose the range of robot orientations \([\theta, \theta']\) into non-overlapping intervals

\[ I_u = [\gamma_u, \gamma_{u+1}], \quad \kappa^u = [x_\kappa, x'_\kappa] \times [y_\kappa, y'_\kappa] \times I_u \]

2. Compute \(CB_j[k^u]\) and \(CT_i[k^u]\), then \(RB_j[k^u]\) and \(R'T_i[k^u]\)

3. Obtain void cells decomposition \(K_{\text{void}}\) of void configuration space \(C^u_{\text{void}}\)

\[
C^u_{\text{void}} = \kappa^u \setminus \left\{ \bigcup_{i=1}^{n} \mathcal{RB}_j[k^u] \right\} \\
\quad \quad \quad \quad \quad \quad \quad \quad \cup \bigcup_{i=1}^{r} \mathcal{R'T}_i[k^u] \}
\]

4. Obtain observation cells decomposition \(K_z\) of \(C^u_z\)

\[
C^u_z = \bigcup_{i=1}^{r} \mathcal{R'T}_i[k^u] \setminus \bigcup_{j=1}^{n} \mathcal{RB}_j[k^u]
\]
Influence of Sensor Geometry

Sensor geometry $S_1$

<table>
<thead>
<tr>
<th>Path</th>
<th>Targets</th>
<th>$D_{tot}$</th>
<th>$B_{tot}$</th>
<th>$\eta_{CL}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau^*$</td>
<td>$T_1, T_4, T_5$</td>
<td>21.3</td>
<td>14.4</td>
<td>0.0516</td>
</tr>
<tr>
<td>$\tau_1$</td>
<td>$T_1, T_2, T_3$</td>
<td>22.6</td>
<td>14.6</td>
<td>0.0381</td>
</tr>
</tbody>
</table>

Sensor geometry $S_2$

<table>
<thead>
<tr>
<th>Path</th>
<th>Targets</th>
<th>$D_{tot}$</th>
<th>$B_{tot}$</th>
<th>$\eta_{CL}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau^*$</td>
<td>$T_1, T_2, T_3$</td>
<td>22.6</td>
<td>14.6</td>
<td>0.0381</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>$T_4, T_5$</td>
<td>22.6</td>
<td>14.6</td>
<td>0.0226</td>
</tr>
</tbody>
</table>
Large Workspace (Information Roadmap)
## Performance Comparison

<table>
<thead>
<tr>
<th>Efficiency Metric</th>
<th>Method</th>
<th>Optimal Strategy, $\sigma^*$</th>
<th>Shortest Path ($\sigma^*$ Improvement)</th>
<th>Complete Coverage ($\sigma^*$ Improvement)</th>
<th>Random Search ($\sigma^*$ Improvement)</th>
<th>Grid Search ($\sigma^*$ Improvement)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\eta_W$</td>
<td></td>
<td>0.4610</td>
<td>0.3053 (51.0%)</td>
<td>0.2683 (71.8%)</td>
<td>0.1441 (219.9%)</td>
<td>0.2321 (98.6%)</td>
</tr>
<tr>
<td>$\eta_Y$</td>
<td></td>
<td>0.0595</td>
<td>0.0407 (46.2%)</td>
<td>0.0055 (981.8%)</td>
<td>0.0114 (421.9%)</td>
<td>0.0122 (387.7%)</td>
</tr>
<tr>
<td>$\eta_{\mathcal{L}}$</td>
<td></td>
<td>0.0446</td>
<td>0.0157 (184.1%)</td>
<td>0.0153 (191.5%)</td>
<td>0.0098 (355.1%)</td>
<td>0.0133 (235.3%)</td>
</tr>
<tr>
<td>$\eta_H$</td>
<td></td>
<td>0.0599</td>
<td>0.0330 (81.5%)</td>
<td>0.0410 (46.1%)</td>
<td>0.0244 (145.5%)</td>
<td>0.0343 (74.6%)</td>
</tr>
</tbody>
</table>

Given a set $\mathcal{P}$ of $N$ pursuers and a set $\mathcal{T}$ of $M$ targets moving within an obstacle-populated game area $S$, find a set of policies which maximize the total sensing reward, and minimize the total time required to capture targets in $\mathcal{T}$ that have been positively detected.

**Assumptions**
1. Targets travel in straight lines with constant velocity
2. Targets are observed intermittently by multiple sensors that measure only position
3. Pursuers have two modes: detection and pursuit
4. Tracks may be unobserved, partially-observed ($< k$), or fully-observed ($\geq k$)
5. Pursuers can always move faster than the targets

**Objectives**
- Maximize the probability of detecting unobserved tracks
- Maximize the probability of detecting partially-observed tracks
- Minimize the distance traveled to detect and capture targets
Example of cell decomposition

- Rectangular workspace \((L_1 \times L_2)\)
- Four C-obstacles
- One target with \(2 < k\) detections
- One sensor with range \(r\)

Connectivity Graph

Obstacle free cells

Observation cells
THEOREM 4.1. The pursuit-evasion game in Problem 2.1 is guaranteed to terminate provided,

\[ N \geq N_{\text{min}} = \frac{1}{2} \left[ \left\lfloor \frac{2L}{r} \right\rfloor + k - 1 + \left\lfloor \frac{2L}{r} \right\rfloor - k + 3 \right] \]  \hspace{1cm} (4.1)

and requires a time,

\[ t_f \leq T_u = \frac{(\sqrt{2}L - 2r)}{V_{\tau_{\text{min}}}} + \left\lfloor \frac{(k - 2)M}{N} \right\rfloor + 1 \frac{(\sqrt{2}L - r)}{V_p} \]

\[ + \frac{r}{(V_{p_{\text{max}}}^2 - V_{\tau}^2)} + \left\lfloor \frac{M}{N} \right\rfloor \frac{(V_{\tau} + \sqrt{2V_{p_{\text{max}}}^2} - V_{\tau})}{(V_{p_{\text{max}}}^2 - V_{\tau}^2)^2} L \]  \hspace{1cm} (4.3)

to capture all \( M \) targets in \( T \). If the network contains at least

\[ N_r = \frac{1}{2} \left[ \ell \left\lfloor \frac{2L}{r} \right\rfloor - 4\ell(\ell - 1) + (k - 2)M + \ell \left\lfloor \frac{2L}{r} \right\rfloor - 4\ell(\ell - 1) - (k - 2)M \right] \]  \hspace{1cm} (4.4)

sensors, with \( \ell = 1, \ldots, \lfloor L/4r \rfloor \), then all targets in \( T \) can be captured in a time,

\[ t_f \leq T_\ell = \frac{1}{V_{\tau_{\text{min}}}} \left\{ \frac{\sqrt{2}}{2}L - 2\sqrt{2}r(\ell - 1) + 2r[1 + \sqrt{2}(\ell - 1)] - \frac{\sqrt{2}}{2}L \right\} \]

\[ + \frac{(\sqrt{2}L - r)}{V_p} + \frac{r}{(V_{p_{\text{max}}}^2 - V_{\tau}^2)} \]  \hspace{1cm} (4.5)

and the game terminates in \( t_f \leq T_\ell \leq T_u \), where \( T_\ell = T_u \) when \( \ell = 1 \) and \( k = 3 \).
Heterogeneous Network

Ground sensor

Air sensor
Heterogeneous Network

C-target

Undetected target track
Heterogeneous Network

Partially-observed track
Heterogeneous Network

Ground sensor in pursuit mode

Captured target
Experiments

-Conducted by Prof. Rafael Fierro and Brent Perteet, University of New Mexico
Conclusions

- Geometric and probabilistic sensor models
- Track Coverage Functions
- Information Value Functions
- Optimal Control of Cooperative Sensor Networks
- Underwater, ground, and air robots

Work in progress:
- Maneuvering targets
- Path Exposure
- Online Learning and Fusion
- Optimal Control of Distributions
LISC Research

Active Research Areas
Approximate dynamic programming
Adaptive control of aircraft
Learning
Artificial and spiking neural networks
Games (CLUE®, Marco Polo, Pacman®)

Acknowledgments

Collaborators:
Dr. Thomas Wettergren, Naval Undersea Warfare Center
Prof. Rafael Fierro, University of New Mexico
Prof. Anil Rao, University of Florida

Sponsors:
This research is funded by the ONR (Code 321), NASA, and the NSF ECS Division
Back-up Slides
Energy Expenditure

Legend:
- Black: Cooperative optimal control ($x_0^*$)
- Red: Non-cooperative path planning, given $x_0$ and $x_f$
## Comparison with Other Deployments

<table>
<thead>
<tr>
<th>Mission Parameters</th>
<th>Performance Metrics</th>
<th>Optimal Control &amp; $x_0^*$</th>
<th>Optimal Control</th>
<th>Path Planning</th>
<th>Optimal Buoys</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(n, k) = (15, 3)$</td>
<td>Track Coverage</td>
<td>$2.52 \cdot 10^4$</td>
<td>$1.74 \cdot 10^4$</td>
<td>$8.87 \cdot 10^3$</td>
<td>$1.99 \cdot 10^3$</td>
</tr>
<tr>
<td>$\Delta T = 3$ days</td>
<td>Energy</td>
<td>$165$</td>
<td>$1.31 \cdot 10^2$</td>
<td>$604$</td>
<td>$0$</td>
</tr>
<tr>
<td></td>
<td>Total Performance</td>
<td>$2.50 \cdot 10^4$</td>
<td>$1.73 \cdot 10^4$</td>
<td>$8.27 \cdot 10^3$</td>
<td>$1.99 \cdot 10^3$</td>
</tr>
<tr>
<td></td>
<td>OC* Improvement</td>
<td>n/a</td>
<td>44%</td>
<td>202.3%</td>
<td>1,156 %</td>
</tr>
</tbody>
</table>

**Where:**

- **Optimal Control & $x_0^*$ (OC*):** Sensor network's initial positions, control and state histories are optimized simultaneously.
- **Optimal Control:** Sensor network's initial positions are given, and the control and state histories are optimized simultaneously.
- **Path Planning:** Sensor network's initial and final position are optimized with respect to $SS$, and the control and state histories are optimized with respect to $Energy$.
- **Zero Control:** Sensor network's initial positions are optimized with respect to the currents and $SS$, but the sensors have no on-board controls (e.g., buoys).
The decision tree $DT$ obtained from $T_r$ is a tuple $\{U_C, U_D, R, A\}$ with $\kappa_0$ as the root, and the value of reward function $R$ as the leaves. Where,

- $U_C$: set of chance nodes (round);
- $U_D$: set of test-decision nodes (squares);
- $A$: directed arcs.

The **optimal path** in the decision tree is found using a rolling-back procedure that determines the optimal strategy by recursively estimating the utility of each branch.

The connectivity tree $T_r$ associated with $G$ and two cells $\kappa_0 \ni q_0$, $\kappa_f \ni q_f$, is a tree graph with $\kappa_0$ as the root, $\kappa_f$ as the leaves, a cost $d$ attached to each arc, and with the following properties:

- A branch $\tau$ in $T_r$ represents a channel joining $\kappa_0$ to $\kappa_f$ in $G$.
- Two branches are said to be information equivalent if they join the same cells, $\kappa_i$ and $\kappa_j$, and contain the same set of observation cells, regardless of the order.
- A branch in $T_r$ connecting any two cells $\kappa_i$ and $\kappa_j$ has the smallest overall cost of any other information-equivalent branch in $G$.

**Label-correcting pruning algorithm:**

**Connectivity graph, $G$**

**Connectivity tree, $T_r$**
Performance Analysis

**Performance of pruning algorithm:**

<table>
<thead>
<tr>
<th>Method</th>
<th>Nodes</th>
<th>Arcs in $G$</th>
<th>Observation cells in $G$</th>
<th>Branches in $T_r$</th>
<th>Computation time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pruning Algorithm</td>
<td>531</td>
<td>686</td>
<td>270</td>
<td>48</td>
<td>79 s</td>
</tr>
<tr>
<td>Exhaustive Search</td>
<td>51</td>
<td>106</td>
<td>43</td>
<td>51</td>
<td>87 s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>$d_M$</th>
<th>Branches in $T_r$</th>
<th>Time slices in $T_r$</th>
<th>Computation time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pruning Algorithm</td>
<td>None</td>
<td>16</td>
<td>17</td>
<td>7 s</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>8</td>
<td>15</td>
<td>3 s</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4</td>
<td>9</td>
<td>1 s</td>
</tr>
<tr>
<td>Exhaustive Search</td>
<td>None</td>
<td>34835</td>
<td>19</td>
<td>1570 s</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>27452</td>
<td>19</td>
<td>1064 s</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>13693</td>
<td>17</td>
<td>378 s</td>
</tr>
</tbody>
</table>

**Complexity of decomposition:** $O(n_T(n_B + n_T)) + O((n_B + n_T) \log (n_B + n_T))$

$n_\mathcal{A},$ # of edges defining robot $\mathcal{A};\ n_B,$ # of total edges of $n$ convex obstacles; $n_T,$ # of total edges of $r$ convex targets

Proportional Controller: 

\[ u_p^i = K_p \zeta_i \]

where \( \zeta_i = [\ell_i \ \psi_i]^T \) is the error vector and 

\[ K_p = \text{diag}(k_v, k_\omega) \] is the diagonal control constant matrix
Interception point is computed using Newton's method

\[ \delta(t_c) = \begin{bmatrix} x_t + t_c v_t \cos \theta_t \\ y_t + t_c v_t \sin \theta_t \end{bmatrix} \]

\[ t_c = \frac{r \psi + \| c - \delta(t_c) \| \cos \alpha}{v_p} \]

\[ c = \begin{cases} c_R, & \text{if } \| c_R - \delta \| \leq \| c_L - \delta \|, \\ c_L, & \text{if } \| c_R - \delta \| > \| c_L - \delta \|. \end{cases} \]

## Target Tracks

1. **An unobserved track** is the path of a target \( j \) for which there are no detections at the present time, \( t \).

2. **A partially-observed track** is the path of a target that is estimated from \( 0 < l < k \) individual sensor detections obtained up to the present time, \( t \).

3. **A fully-observed track** is the path of a target that is estimated from at least \( k \) individual sensor detections obtained up to the present time, \( t \).

## Cells

1. **A void cell** is a convex polygon \( \kappa \subset C_{free} \) with the property that for every configuration \( q_i \in \kappa \) the pursuer \( i \) has zero probability of detecting a partially-observed target.

2. **An observation cell** is a convex polygon \( \kappa \subset C_{free} \) with the property that for every configuration \( q_i \in \kappa \) the pursuer \( i \) has a non-zero probability of detecting a partially-observed target.
Control Policy for Sensors in Detection Mode

Reward function:

\[ R(\kappa_l, \kappa_r) = w_1 P_d(\kappa_l) + w_2 \Delta P^k_S(\kappa_l, \kappa_r) - w_3 d(\kappa_l, \kappa_r) \]

where \( \Delta P^k_S(\kappa_l, \kappa_r) \) is the change in the network track-coverage,

\( d(\kappa_l, \kappa_r) \) is the Euclidean distance between cells,

\( P_d(\kappa_l) \) is the probability of detecting a target inside a given cell (assuming a binary sensor model), and

\( w_1, w_2, \text{ and } w_3 \) are weighting parameters.

Using a graph searching algorithm such as A*, the optimal sequence or channel of cells which maximizes the reward is

\[ \mu^* \equiv \{\kappa_0, \ldots, \kappa_f\}^* = \arg \max_{\mu} \sum_{(\kappa_l, \kappa_r) \in \mu} R(\kappa_l, \kappa_r) \]
Results: 5 Sensors and 1 Target

- Detection points
- Static sensors
- Target
- Mobile sensor
- Hypothesized track
- Initial location of mobile sensor
- Optimal path
- Obstacles

Results: 5 Sensors and 1 Target
Results: 5 Sensors and 1 Target
Results: 5 Sensors and 2 Targets
CLUE® is a benchmark example of treasure-hunt problem, because the information (or evidence) that can be obtained about the hidden variable, depends on the position of the pawn on the gameboard: coupled motion planning and inference problems.

**Game characteristics:**
- “who, how, and what room?”
- 6 suspects, 6 weapons, 9 rooms
- Movement
- Suggestion decision → evidence
- Inference of hidden cards
- CLUE® ↔ surveillance systems

Connectivity tree, \( T \), is folded into an influence diagram (action decisions, \( a_k \), observable state, \( x_k \))

The observation cells in \( \Omega(x_k) \) specify the admissible set of test decisions, \( u_k \), and the domain of the non-observable state, \( \Omega(z_k) = \{ m_i, \ldots, m_j \} \)

\( Z_T = \{ z_1, z_2, \ldots, z_{f-1} \} \) a sequence of measurements about \( y \) over \( \{ t_1, t_2, \ldots, t_f \} \)

**Profit of Observation:** \( v(t_k) = R(t_k) = w_B . B(t_k) - w_J . J(t_k) - w_D . D(t_k) \)

where \( B(t_k) \) is the expected entropy reduction (EER),

\[ \Delta H(t_k) = H(y \mid z_{k-1}, z_{k-2}, \ldots, z_1) - H(y \mid z_k, z_{k-1}, \ldots, z_1) = I(y; z_k \mid z_{k-1}, \ldots, z_1) \]
Optimal CLUE® Game Strategy

**Profit of Observation:**
\[ R(t_k) = w_B B(t_k) - w_J J(t_k) - D(t_k) \]

- ICP
- \( q_0 = 66 \)
- \( q_f = 64 \)

**Shortest path** \((w_B = 0; w_J = 0)\):
\[ [66 \ 63 \ 64]; \quad R = -7; \]

**Aggressive path** \((w_B = 12)\):
\[ [66 \ 63 \ 62 \ d \ 62 \ 63 \ 64]; \quad R = 12 \times 0.41 - 0 - 7 = -2.08; \]

**Suggestion:**
- \{Mrs. Peacock; Revolver; Dining Room\}

**P3’s Response:** \{Revolver\}

**Posterior before evidence:**
\[
\begin{bmatrix}
0.25 & 0.25 & 0.25 & 0.25 \\
0.25 & 0.25 & 0.25 & 0.25 \\
0.14 & 0.14 & 0.14 & 0.14 & 0.14 & 0.14 & 0.14
\end{bmatrix}
\]

**Posterior after evidence:**
\[
\begin{bmatrix}
0.5 & 0.17 & 0.17 & 0.17 & 0.17 \\
0.33 & 0.33 & 0 & 0 & 0.33 \\
0.13 & 0.13 & 0.13 & 0.13 & 0.25 & 0.13 & 0.13
\end{bmatrix}
\]

**Suggestion:**
- \{Mr. Green; Rope; Lounge\}

**P2’s Response:** \{Mr. Green\}
# Game Results

<table>
<thead>
<tr>
<th>Players:</th>
<th>Games won / games played</th>
<th>Winning rate</th>
<th>Time to determine $y^*$</th>
<th>Time to win the game</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID player</td>
<td>14 / 25</td>
<td>56 %</td>
<td>8.21 turns</td>
<td>8.57 turns</td>
</tr>
<tr>
<td>Humans</td>
<td>10 / 25</td>
<td>40 %</td>
<td>n.a.</td>
<td>12.7 turns</td>
</tr>
<tr>
<td>CS player</td>
<td>1 / 25</td>
<td>4 %</td>
<td>12 turns</td>
<td>12 turns</td>
</tr>
<tr>
<td>BN player</td>
<td>20 / 37</td>
<td>54 %</td>
<td>10.2 turns</td>
<td>10.9 turns</td>
</tr>
<tr>
<td>Humans</td>
<td>16 / 37</td>
<td>43 %</td>
<td>n.a.</td>
<td>12.9 turns</td>
</tr>
<tr>
<td>CS player</td>
<td>1 / 37</td>
<td>2.7 %</td>
<td>4 turns</td>
<td>4 turns</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Players:</th>
<th>Games won / games played</th>
<th>Winning rate</th>
<th>Time to determine $y^*$</th>
<th>Time to win the game</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID player</td>
<td>21 / 43</td>
<td>48.8 %</td>
<td>10.5 turns</td>
<td>10.7 turns</td>
</tr>
<tr>
<td>Humans</td>
<td>19 / 43</td>
<td>44 %</td>
<td>n.a.</td>
<td>8.89 turns</td>
</tr>
<tr>
<td>CS player</td>
<td>3 / 43</td>
<td>6.98 %</td>
<td>8.33 turns</td>
<td>8.33 turns</td>
</tr>
<tr>
<td>BN player</td>
<td>18 / 55</td>
<td>32.7 %</td>
<td>11.7 turns</td>
<td>12.0 turns</td>
</tr>
<tr>
<td>Humans</td>
<td>32 / 55</td>
<td>58 %</td>
<td>n.a.</td>
<td>11.4 turns</td>
</tr>
<tr>
<td>CS player</td>
<td>5 / 55</td>
<td>9 %</td>
<td>10.6 turns</td>
<td>10.6 turns</td>
</tr>
</tbody>
</table>