Michigan-AFRL Collaborative Center in Control Science

Semi-Annual Review
Cooperative surveillance and pursuit

Jonathan Las Fargeas

Department of Aerospace Engineering
University of Michigan
Ann Arbor, Michigan, USA

April 22nd, 2013
Motivation

Persistent ISR missions
- Multiple unmanned vehicles patrolling an area.
- Multiple unattended sensors

Base defense missions
- Friendly base also located in area.
- Intruders attempting to reach base.

1 U.S. Air Force photo by Bobbi Zapka
2 U.S. Army Spc. Kristina Gupton / Wikimedia Commons / CC-BY-SA-2.0
*Enabling a Team of UAVs to Cooperatively Track Intruders on a Road Network. Dr. Derek Kingston, Control Science Center of Excellence, Air Vehicles Directorate, Air Force Research Laboratory, April 19, 2012.
Research Objective

Global objective

- Autonomously track state of objects of interest in theater of operations over time.

Year 1 (2011-2012)

- Persistent Visitation
- Persistent Visitation with Fuel Constraints
- Cooperative surveillance and pursuit using UAVs and UGSs

Year 2 (2012-2013)

- Selection of locations and revisit deadlines of unattended sensors
Many manned and unmanned Air Force platforms perform persistent ISR missions.

These platforms collect large amounts of data which require many operators to analyze before classification decisions can be made and action taken.

Automating part of this data analysis to reduce human workload is a priority.
Air Force Relevance
Degraded Communications

- These platforms can assist with communications if need be:

“Persistent near-space systems, potentially in the form of ultra-long endurance airships or autonomous flight vehicles, can ensure theater-level communications relay functions in the event of loss or degradation of corresponding space systems.”

-Technology Horizons Report

Air Force Relevance

Increased autonomy

These platforms can perform a wide range of functions:

- Autonomously decide how to pursue intruder
  - Faster reactions in time-critical situations
- Autonomously image intruder
- Autonomously decide placement of unattended sensors
- Autonomously decide when to refuel

<table>
<thead>
<tr>
<th>Problem</th>
<th>Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patrolling</td>
<td>Persistent Coverage [1]</td>
</tr>
<tr>
<td></td>
<td>Minimization of uncertainty [2],[3]</td>
</tr>
<tr>
<td></td>
<td>Frequency-based [4],[5],[6]</td>
</tr>
<tr>
<td>Pursuit-Evasion</td>
<td>Games [7],[8],[9]</td>
</tr>
<tr>
<td></td>
<td>Patrolling-Security [10],[11]</td>
</tr>
<tr>
<td></td>
<td>Intruder state estimation [12]</td>
</tr>
<tr>
<td>Optimal Sensor Placement</td>
<td>Minimum cut-sets [13]</td>
</tr>
<tr>
<td></td>
<td>Genetic Algorithms [14],[15]</td>
</tr>
<tr>
<td></td>
<td>Stochastic programming [16]</td>
</tr>
</tbody>
</table>
Contributions

⇒ Posed a patrolling problem without a priori constraints on periodicity and proved existence of periodic paths.
⇒ Provided algorithm to find path when including fuel constraints.
⇒ Provided algorithm for multiple UAVs relying on UGSs to patrol and pursue an intruder in a base defense scenario.
⇒ Give methods to place unattended sensors and select their revisit deadlines in surveillance scenarios.

Impacts of research

- Two accepted, one submitted conference papers.
- Collaborative research with AFRL during summer 2012.
Outline

- Motivation & Air Force Relevance
  - **UAV path planning**
    - Persistent Visitation
    - Cooperative Surveillance and Pursuit using UAVs and UGSs
  - Problem statement for selection of sensors’ locations and revisit deadlines
- Model
- Placing sensors
  - At road intersections
  - On roads
  - At road intersections and on roads
- Selecting revisit deadlines
- Conclusion & Future work
Number of objects with unknown characteristics in an area.

Each object has a revisit deadline which if satisfied leads to proper classification.

A UAV is tasked with persistently visiting the objects such that their revisit deadlines are met.

Goal is to find UAV paths which satisfy revisit deadlines.

*Persistent Visitation under Revisit Constraints. ICUAS 2013. Accepted.*
UAV Path Planning
Persistent Visitation under Revisit Constraints

⇒ Proved the existence of periodic paths for the UAV.
⇒ Demonstrated the problem to be NP-complete.

*Persistent Visitation under Revisit Constraints. ICUAS 2013. Accepted.*
Heuristics used to solve the problem:

- Earliest deadline first
- Search ahead heuristics
  - Maximum minimum slack time
  - Maximum sum of slack times
Degraded communications

- Each object of interest is a sensor generating data at a certain rate.
- Revisit deadline is time till sensor memory is full.
- UAVs used to relay information.
UAV Path Planning
Persistent Visitation with Fuel Constraints

- Included fuel constraints and multiple fuel stations with varying costs for fuel.
- Goal is to find a minimal cost cycle which satisfies the revisit deadlines.

⇒ Provided algorithm to compute minimal cost path for vehicle.

*Persistent Visitation with Fuel Constraints. EWGT 2012.*
UAV Path Planning
Persistent Visitation with Fuel Constraints

- Included fuel constraints and multiple fuel stations with varying costs for fuel.
- Goal is to find a minimal cost cycle which satisfies the revisit deadlines.

⇒ Provided algorithm to compute minimal cost path for vehicle.

```plaintext
Path Planning
Persistent Visitation with Fuel Constraints

- Included fuel constraints and multiple fuel stations with varying costs for fuel.
- Goal is to find a minimal cost cycle which satisfies the revisit deadlines.

⇒ Provided algorithm to compute minimal cost path for vehicle.

*Persistent Visitation with Fuel Constraints. EWGT 2012.*
UAV Path Planning
Surveillance and Pursuit using Unattended Ground Sensors

- Base defense scenario with multiple UAVs without ATR capabilities.
- Intruder attempts to reach base via road network.
- UGSs placed along road network with revisit deadlines set by mission designer.
- UAVs must image intruders before base is reached while minimizing deadlines missed.

http://www.youtube.com/v/BTfr2VAV_bM?rel=0
Humans not needed to analyze images ⇒ Reduced human workload

Currently working on implementing these algorithms to interface with AFRL software for potential flight demonstrations.
Humans not needed to analyze images ⇒ Reduced human workload

Currently working on implementing these algorithms to interface with AFRL software for potential flight demonstrations.
Outline

- Motivation & Air Force Relevance
- UAV path planning
  - Persistent Visitation
  - Cooperative Surveillance and Pursuit using UAVs and UGSs

⇒ Problem statement for selection of sensors’ locations and revisit deadlines

⇒ Model
- Placing sensors
  - At road intersections
  - On roads
    - At road intersections and on roads
- Selecting revisit deadlines
- Conclusion & Future work
Problem Statement

Previous Assumptions
- ‘Good’ UGS locations chosen by mission designer.
- ‘Good’ revisit deadlines selected by mission designer.

Remove assumptions ⇒ New questions
- Where should UGSs be placed?
- How should revisit deadlines be selected?

Problem
- Place UGSs and set their revisit deadlines to improve the defenders’ detection capabilities.
- Abstract UAV movements into revisit deadlines.
- Assume set of UGS locations and revisit deadlines is flyable.
- Model road network as a weighted digraph $G = (N, E, L)$ where a node $n \in N$ is a road intersection and an edge $e \in E$ is a road.
Intruders

- Assume intruder has a finite set of paths available along the road network.
- Defenders have an estimate of the probability distribution for potential intruder movements in the road network, $p_{i,j}$.
The probability distribution for potential intruder movements can acquired through multiple means:

- Deduced from existing intelligence
- Known a priori
- Derived from model of intruder movements
  - Markov process: use steady-state distribution
- Red team
  - Currently working with David Cho, another ARCLAB student, on optimal intruder strategies using a game theoretic approach.
Outline

- Motivation & Air Force Relevance
- UAV path planning
  - Persistent Visitation
  - Cooperative Surveillance and Pursuit using UAVs and UGSs
- Problem statement for selection of sensors’ locations and revisit deadlines
- Model
  ⇒ **Placing sensors**
  - At road intersections
  - On roads
  - At road intersections and on roads
- Selecting revisit deadlines
- Conclusion & Future work
Placing Perfect Sensors

At road intersections

Goal: Maximize number of detections

- $g$: number of available UGSs
- UGSs placed at road intersections when $0 < g \leq |N|

  - $u \in \{0, 1\}^{|N|}$: vector indicating UGSs placement.
  - Approximate $p_{i,j}$ into probability of intruder presence at nodes $p_i, 1 \leq i \leq |N|$.

$$p_i = \sum_{j=1}^{|N|} p_{j,i}$$

$$\max_u (u^T \cdot p) \text{ s.t. } u^T \cdot J_{1 \times |N|} = g$$

![Diagram showing network of nodes and probability values]
Placing Perfect Sensors

At road intersections

Goal: Maximize number of detections

- $g$: number of available UGSs
- UGSs placed at road intersections when $0 < g \leq |N|
  - $u \in \{0, 1\}^{|N|}$: vector indicating UGSs placement.
  - Approximate $p_{i,j}$ into probability of intruder presence at nodes $p_i$, $1 \leq i \leq |N|$.

\[
p_i = \sum_{j=1}^{|N|} p_{j,i}
\]

\[
\max_u (u^T \cdot p) \quad \text{s.t.} \quad u^T \cdot J_{1 \times |N|} = g
\]
Placing Perfect Sensors
At road intersections

Goal: Maximize number of detections

- $g$ : number of available UGSs
- UGSs placed at road intersections when $0 < g \leq |N|$
  - $u \in \{0, 1\}^{N}$ : vector indicating UGSs placement.
  - Approximate $p_{i,j}$ into probability of intruder presence at nodes $p_i$, $1 \leq i \leq |N|$.

$$p_i = \sum_{j=1}^{N} p_{j,i}$$

$$\max_u (u^T \cdot p) \text{ s.t. } u^T \cdot J_{1 \times |N|} = g$$
Placing Perfect Sensors

On roads

Goal: Maximize number of detections

- UGSs placed at beginning of roads
- UGSs placed along roads when $|E| > g > |N|$

$U \in \{0, 1\}^{N \times N}$: matrix indicating UGSs placement.

\[
P = \begin{bmatrix}
0 & p_{12} & \ldots & p_{1n} \\
p_{21} & 0 & \ldots & p_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
p_{n1} & p_{n2} & \ldots & 0
\end{bmatrix}
\]

\[
\max_U \text{tr}(P \cdot U^T) \text{ s.t. } \text{tr}(U \cdot J_{|N|}) = g
\]
Placing Perfect Sensors

On roads

Goal: Maximize number of detections

- UGSs placed at beginning of roads
- UGSs placed along roads when $|E| > g > |N|$

$U \in \{0, 1\}^{N \times |N|}$: matrix indicating UGSs placement.

$$P = \begin{bmatrix}
0 & p_{12} & \cdots & p_{1n} \\
p_{21} & 0 & \cdots & p_{2n} \\
\vdots \\
p_{n1} & p_{n2} & \cdots & 0
\end{bmatrix}$$

$$\max_U \text{tr}(P \cdot U^T) \text{ s.t. } \text{tr}(U \cdot J_{|N|}) = g$$
Placing Perfect Sensors

At road intersections and on roads

Goal: Maximize utility of UGSs

- **Utility**: quality of an UGS detection
  - How well the intruder’s position can be predicted given the detection
    - Node: detect intruders more often, have an idea of where intruder may be
    - Edge: detect intruders less often, know exactly where intruder is
  - \[ \Downarrow \]
    - Node: higher frequency, lower quality
    - Edge: lower frequency, higher quality

- **Node utility**: \( V_{i,i} \propto \) weighted average of length of possible outcome edges
- **Edge utility**: \( V_{i,j} \propto \) edge length
Placing Perfect Sensors
At road intersections and on roads

Goal: Maximize utility of UGSs

- Utility: quality of an UGS detection
  - How well the intruder’s position can be predicted given the detection
    - Node: detect intruders more often, have an idea of where intruder may be
    - Edge: detect intruders less often, know exactly where intruder is

\[ \Downarrow \]

- Node: higher frequency, lower quality
- Edge: lower frequency, higher quality

- Node utility: \( V_{i,i} \propto \text{weighted average of length of possible outcome edges} \)
- Edge utility: \( V_{i,j} \propto \text{edge length} \)
Placing Perfect Sensors
At road intersections and on roads

Goal: Maximize utility of UGSs

- Utility: quality of an UGS detection
  - How well the intruder’s position can be predicted given the detection
    - Node: detect intruders more often, have an idea of where intruder may be
    - Edge: detect intruders less often, know exactly where intruder is

  - Node: higher frequency, lower quality
  - Edge: lower frequency, higher quality

  - Node utility: \( V_{i,i} \propto \) weighted average of length of possible outcome edges
  - Edge utility: \( V_{i,j} \propto \) edge length
Placing Perfect Sensors
At road intersections and on roads
Goal: Maximize utility of UGSs

- Utility: quality of an UGS detection
  - How well the intruder’s position can be predicted given the detection
    - Node: detect intruders more often, have an idea of where intruder may be
    - Edge: detect intruders less often, know exactly where intruder is
      ▼
      - Node: higher frequency, lower quality
      - Edge: lower frequency, higher quality

- Node utility: \( V_{i,i} \propto \) weighted average of length of possible outcome edges
- Edge utility: \( V_{i,j} \propto \) edge length
Placing Perfect Sensors
At road intersections and on roads

Goal: Maximize utility of UGSs

$$\max_U \left( \sum_{i=1}^{N} \sum_{j=1}^{N} V_{i,j} \cdot U_{i,j} \right) \text{ s.t. } tr(U \cdot J_{|N|}) \leq g$$

- Finding solutions
  - Search space for $U$ is very large
  - $\frac{(|N|+|E|)!}{(|N|+|E|-g)!}$ candidate solutions
  - Use k-opt approach by switching 0s and 1s
  - Reduce number of candidates to $\frac{g!}{(g-k)!} \times \frac{(|N|+|E|-g)!}{(|N|+|E|-g-k)!}$
Placing Perfect Sensors
At road intersections and on roads

Goal: Maximize utility of UGSs

\[
\max_U (\sum_{i=1}^{N} \sum_{j=1}^{N} V_{i,j} \cdot U_{i,j}) \quad \text{s.t.} \quad \text{tr}(U \cdot J_{|N|}) \leq g
\]

- Finding solutions
  - Search space for U is very large
  - \(\frac{|N|+|E|!}{(|N|+|E|−g)!}\) candidate solutions
  - Use k-opt approach by switching 0s and 1s
  - Reduce number of candidates to \(\frac{g!}{(g-k)!} \times \frac{(|N|+|E|−g)!}{(|N|+|E|−g−k)!}\)
Placing Perfect Sensors
At road intersections and on roads

Goal: Maximize utility of UGSs

\[
\max_U \left( \sum_{i=1}^{N} \sum_{j=1}^{N} V_{i,j} \cdot U_{i,j} \right) \quad \text{s.t.} \quad \text{tr}(U \cdot J_{|N|}) \leq g
\]

- Finding solutions
  - Search space for U is very large
  - 93024 candidate solutions
  - Use k-opt approach by switching 0s and 1s
  - Reduce number of candidates to:
    - \( k = 1 \Rightarrow 60 \)
    - \( k = 2 \Rightarrow 2520 \)
Placing Noisy Sensors

- Given knowledge of UGSs’ true positive and true negative rates.

- UGSs placed either at roads intersections or on roads:
  - Minimize probability of missed detections and false alarms.

- UGSs placed at road intersections and on roads
  - Penalize UGSs where missed detections and false alarms are costly.
  - Minimize penalty.
Placing Sensors

⇒ Can use this work to suggest which roads to disable or where checkpoints should be installed.

*Enabling a Team of UAVs to Cooperatively Track Intruders on a Road Network. Dr. Derek Kingston, Control Science Center of Excellence, Air Vehicles Directorate, Air Force Research Laboratory, April 19, 2012.*
Outline

- Motivation & Air Force Relevance
- UAV path planning
  - Persistent Visitation
  - Cooperative Surveillance and Pursuit using UAVs and UGSs
- Problem statement for selection of sensors’ locations and revisit deadlines
- Model
- Placing sensors
  - At road intersections
  - On roads
  - At road intersections and on roads
  ⇒ Selecting revisit deadlines
- Conclusion & Future work
Selecting Revisit Deadlines \( (r_i) \)

- Let \( \bar{p}_i \) be the normalized probability of intruder being at UGS \( i \).
- Want timely detections \( \rightarrow \) have UAVs and intruders visit UGS at similar rates.
- Minimize the Kullback-Leibler divergence between the two distributions: \( \bar{p}_i \) and \( \frac{1}{r_i} \).
- Use Lagrange multiplier:

\[
L = \sum_{i=1}^{g} \ln \left( \frac{\bar{p}_i}{\frac{1}{r_i}} \right) \cdot \bar{p}_i + \lambda \cdot \left( \sum_{i=1}^{g} \frac{1}{r_i} \sum_{j=1}^{g} \frac{1}{r_j} - 1 \right)
\]

\[\downarrow\]

\[
\bar{p}_i \cdot r_i = \bar{p}_j \cdot r_j, \forall i, \forall j \in \{1, 2, \ldots, g\}
\]

Revisit deadlines are proportional to the probability of UGS being visited by an intruder.
Selecting Revisit Deadlines

\[ \bar{p}_i \cdot r_i = \bar{p}_j \cdot r_j, \forall i, \forall j \in \{1, 2, ..., g\} \]

The mission designer then has the choice of setting the expected maximum undetected intruder time,

\[ \sum_{i=1}^{g} \bar{p}_i \cdot r_i = T, \]

or setting a single revisit deadline which would in turn set all of the others.
Selecting Revisit Deadlines

\[ \bar{p}_i \cdot r_i = \bar{p}_j \cdot r_j, \forall i, \forall j \in \{1, 2, \ldots, g\} \]

The mission designer then has the choice of setting the expected maximum undetected intruder time,

\[ \sum_{i=1}^{g} \bar{p}_i \cdot r_i = T, \]

or setting a single revisit deadline which would in turn set all of the others.
Selecting Revisit Deadlines

\[ \bar{p}_i \cdot r_i = \bar{p}_j \cdot r_j, \forall i, \forall j \in \{1, 2, \ldots, g\} \]

The mission designer then has the choice of setting the expected maximum undetected intruder time,

\[ \sum_{i=1}^{g} \bar{p}_i \cdot r_i = T, \]

or setting a single revisit deadline which would in turn set all of the others.
Selecting Revisit Deadlines

Can reproduce revisit deadline selection for $(\hat{p}_i)^k, k \in \mathbb{N}$

- Enables control of revisit schedule’s sensitivity to probability of intruder presence.
  - $k = 0 \Rightarrow$ uniform revisit deadlines
  - $k = 1 \Rightarrow$ revisit deadlines proportional to probability of intruder presence
  - $k > 1 \Rightarrow$ revisit schedule emphasizes likelier nodes
  - $k \to \infty \Rightarrow$ stay at likeliest node
Selecting Revisit Deadlines

Can reproduce revisit deadline selection for \((\hat{p}_i)^k, k \in \mathbb{N}\)

- Enables control of revisit schedule’s sensitivity to probability of intruder presence.
  - \(k = 0 \Rightarrow \) uniform revisit deadlines
  - \(k = 1 \Rightarrow \) revisit deadlines proportional to probability of intruder presence
  - \(k > 1 \Rightarrow \) revisit schedule emphasizes likelier nodes
  - \(k \to \infty \Rightarrow \) stay at likeliest node
Selecting Revisit Deadlines

Can reproduce revisit deadline selection for \((\hat{p}_i)^k, \ k \in \mathbb{N}\)

- Enables control of revisit schedule’s sensitivity to probability of intruder presence.
  - \(k = 0 \Rightarrow\) uniform revisit deadlines
  - \(k = 1 \Rightarrow\) revisit deadlines proportional to probability of intruder presence
  - \(k > 1 \Rightarrow\) revisit schedule emphasizes likelier nodes
  - \(k \to \infty \Rightarrow\) stay at likeliest node
Selecting Revisit Deadlines

Can reproduce revisit deadline selection for $(\hat{p}_i)^k$, $k \in \mathbb{N}$

- Enables control of revisit schedule’s sensitivity to probability of intruder presence.
  - $k = 0 \Rightarrow$ uniform revisit deadlines
  - $k = 1 \Rightarrow$ revisit deadlines proportional to probability of intruder presence
  - $k > 1 \Rightarrow$ revisit schedule emphasizes likelier nodes
  - $k \to \infty \Rightarrow$ stay at likeliest node
Selecting Revisit Deadlines

- Given a limited number of UAVs available for a mission
  ⇒ Revisit schedule may not be flyable
  ⇒ Can use MACCCS work by MIT on inconsistent planning
Outline

- Motivation & Air Force Relevance
- UAV path planning
  - Persistent Visitation
  - Cooperative Surveillance and Pursuit using UAVs and UGSs
- Problem statement for selection of sensors’ locations and revisit deadlines
- Model
- Placing sensors
  - At road intersections
  - On roads
  - At road intersections and on roads
- Selecting revisit deadlines

⇒ Conclusion & Future work
Conclusion & Future work

Conclusion

- Presented UAV path planning methods for
  - Persistent ISR scenarios
  - Base defense scenarios
- Presented methods for placing UGS and setting their revisit deadlines in surveillance scenarios
  ⇒ Reduce human workload
  ⇒ Assist when communications degraded
  ⇒ Increase autonomy
Future Work

- Finish cooperative surveillance and pursuit software for potential AFRL flight demonstrations.
- Examine effectiveness of different sensor timing and placement approaches through simulation.
- Study results’ sensitivity to errors in priors.
- Investigate sensor fusion methods.
- Look into placing UGS such that intruder model can be updated live.
- Investigate larger class of fractionable operations.

Questions?
jfargeas@umich.edu
Placing perfect sensors
At road intersections and on roads

- **Edge Utility:** \( V_{i,j} = p_{i,j} \cdot L_{i,j} \)
- **Node Utility:** \( V_{i,i} \)

- **Number of uncovered outgoing edges from node** \( i \): \( \phi_i \)
  \[
  \phi_i = \sum_{k=1}^{\left| N \right|} (1 - U_{i,k}) \quad \text{where} \quad p_{i,k} > 0
  \]

- **Probability of intruder traveling along uncovered edge:** \( \tilde{p}_{i,j} \)
  \[
  \tilde{p}_{i,j} = \frac{(1-U_{i,j}) \cdot p_{i,j}}{\sum_{k=1}^{\left| N \right|} (1-U_{i,k}) \cdot p_{i,k}}
  \]

\( \Rightarrow \quad V_{i,i} = p_i \sum_{j=1}^{\left| N \right|} L_{i,j} \cdot (\tilde{p}_{i,j})^{\phi_i} \)
Placing perfect sensors
At road intersections and on roads

- **Edge Utility**: \( V_{i,j} = p_{i,j} \cdot L_{i,j} \)
- **Node Utility**: \( V_{i,i} \)

- **Number of uncovered outgoing edges from node** \( i \): \( \phi_i \)
  \[
  \phi_i = \sum_{k=1}^{\left| N \right|} \sum_{k \neq i} (1 - U_{i,k}) \quad \text{if} \quad p_{i,k} > 0
  \]

- **Probability of intruder traveling along uncovered edge**: \( \tilde{p}_{i,j} \)
  \[
  \tilde{p}_{i,j} = \frac{(1-U_{i,j}) \cdot p_{i,j}}{\sum_{k=1}^{\left| N \right|} (1-U_{i,k}) \cdot p_{i,k}}
  \]

\( \Rightarrow V_{i,i} = p_i \sum_{j=1}^{\left| N \right|} L_{i,j} \cdot (\tilde{p}_{i,j})^{\phi_i} \)
Placing perfect sensors
At road intersections and on roads

- Edge Utility: \( V_{i,j} = p_{i,j} \cdot L_{i,j} \)
- Node Utility: \( V_{i,i} \)

Number of uncovered outgoing edges from node \( i \): \( \phi_i \)
\[
\phi_i = \sum_{k=1}^{\mid N \mid} \left( 1 - U_{i,k} \right) \\
\text{subject to} \\
p_{i,k} > 0
\]

Probability of intruder traveling along uncovered edge: \( \tilde{p}_{i,j} \)
\[
\tilde{p}_{i,j} = \frac{(1 - U_{i,j}) \cdot p_{i,j}}{\sum_{k=1}^{\mid N \mid} (1 - U_{i,k}) \cdot p_{i,k}}
\]

\[
\Rightarrow V_{i,i} = p_i \sum_{j=1}^{\mid N \mid} L_{i,j} \cdot (\tilde{p}_{i,j})^{\phi_i}
\]
Placing perfect sensors
At road intersections and on roads

- **Edge Utility:** $V_{i,j} = p_{i,j} \cdot L_{i,j}$
- **Node Utility:** $V_{i,i}$

- **Number of uncovered outgoing edges from node $i$:** $\phi_i$
  
  $\phi_i = \sum_{k=1, k \neq i}^{\left|N\right|} (1 - U_{i,k})$

- **Probability of intruder traveling along uncovered edge:** $\tilde{p}_{i,j}$
  
  $\tilde{p}_{i,j} = \frac{(1-U_{i,j}) \cdot p_{i,j}}{\sum_{k=1}^{\left|N\right|} (1-U_{i,k}) \cdot p_{i,k}}$

  \[ \Rightarrow V_{i,i} = p_i \sum_{j=1}^{\left|N\right|} L_{i,j} \cdot (\tilde{p}_{i,j})^{\phi_i} \]
P.F. Hokayem, D. Stipanovic, and M.W. Spong.  
On persistent coverage control.  

S.L. Smith, M. Schwager, and D. Rus.  
Persistent monitoring of changing environments using a robot with limited range sensing.  

C.G. Cassandras, X.C. Ding, and X. Lin.  
An optimal control approach for the persistent monitoring problem.  
References II

Multi-robot area patrol under frequency constraints.

N. Nigam and I. Kroo.
Persistent surveillance using multiple unmanned air vehicles.

N. Basilico, N. Gatti, and F. Amigoni.
Developing a deterministic patrolling strategy for security agents.
Aigner A. and Fromme M.
A game of cops and robbers.

A. Quilliot.
A short note about pursuit games played on a graph with a given genus.

R. Nowakowski and P. Winkler.
Vertex-to-vertex pursuit in a graph.

N. Basilico, N. Gatti, and F. Amigoni.
Patrolling security games: Definition and algorithms for solving large instances with single patroller and single intruder.


S. Sen, S. Narasimhan, and K. Deb.
Sensor network design of linear processes using genetic algorithms. 

J. A. Carballido, I. Ponzoni, and N. B. Brignole.
CGD-GA: A graph-based genetic algorithm for sensor network design. 

Y. Shastri and U. Diwekar.