

Swarm Intelligence: a New C2 Paradigm with an Application to the Control of Swarms of UAVs

Paolo Gaudiano

**Icosystem Corporation
Cambridge, MA, USA
www.icosystem.com**

Overview

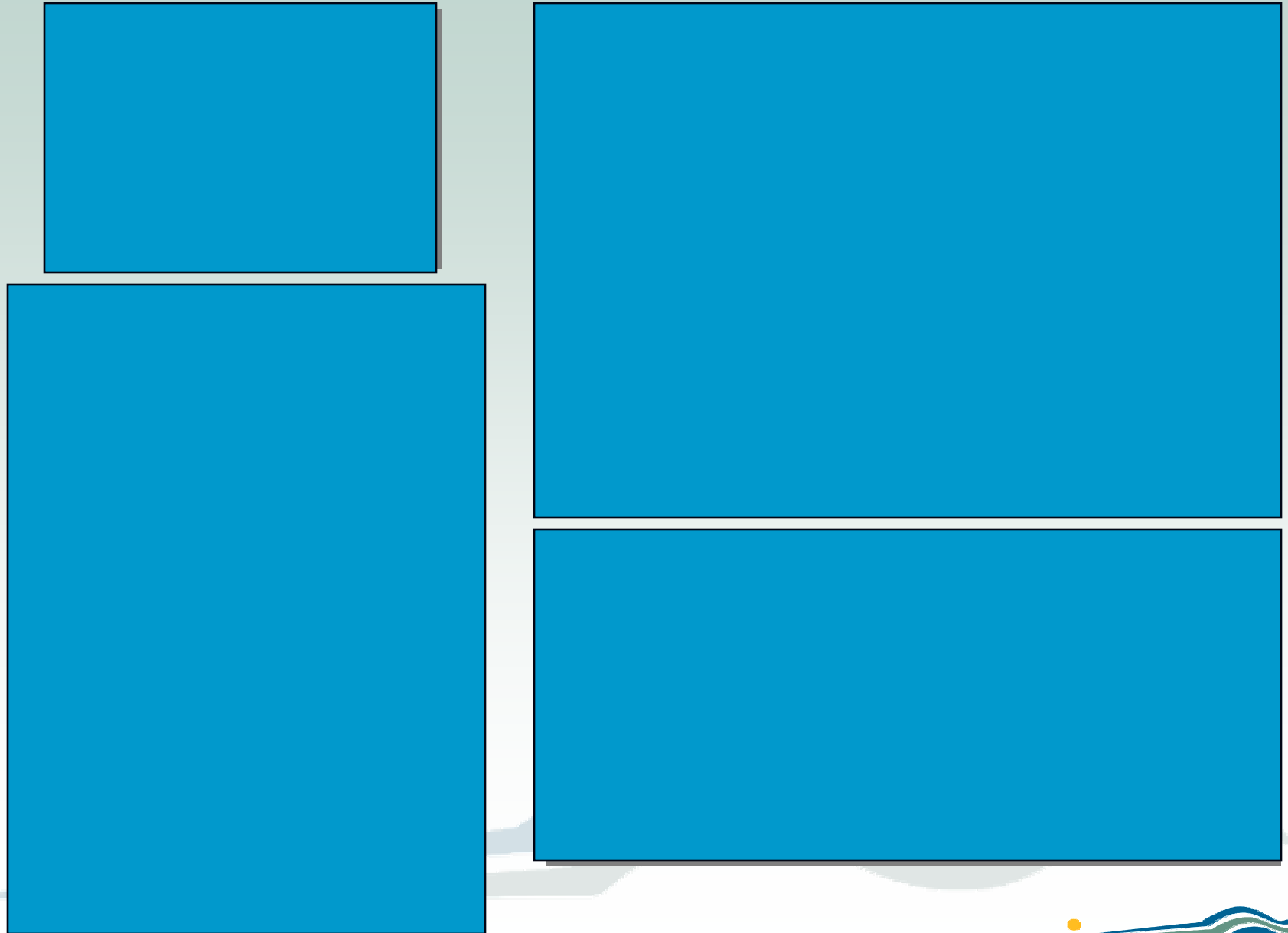
- UAVs: Definition and Examples
- Complex Systems and Swarm Intelligence
- Agent-Based Modeling
- ABM for the control of UAV Swarms
- Conclusions and future work

UAV: Definition

A powered, aerial vehicle that does not carry a human operator, uses aerodynamic forces to provide vehicle lift, can fly autonomously or be piloted remotely, can be expendable or recoverable, and can carry a lethal or non-lethal payload.

Source: DoD UAV Roadmap 2002

Many Types of UAV



Controlling Multiple UAVs

Problem Statement:

- Current UAVs require *at least* one operator per UAV
- Technological advances make multi-UAV missions a near-term reality

Need control strategies that allow one operator to monitor/control multiple UAVs

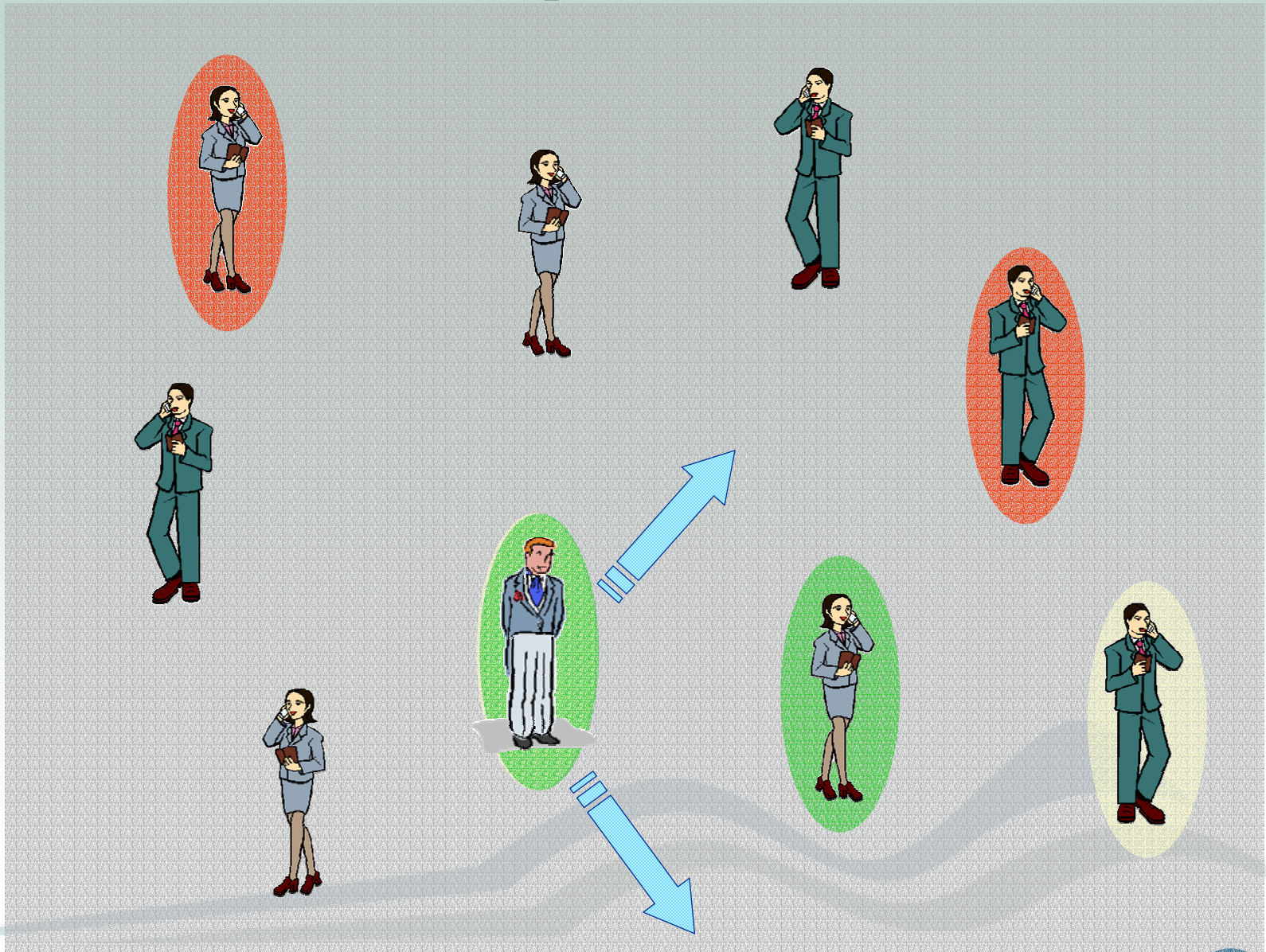
UAV Swarms as Complex Systems

A system is *complex* when:

1. It consists of a large number of elements
2. Significant interactions exist between elements
3. System exhibits *emergent behavior*: cannot predict system behavior from analysis of individual elements

Traditional "reductionist" approaches cannot cope with complex systems

The Icosystem Game



The Icosystem Game

Combinatorial business chemistry

AGGRESSORS - DEFENDERS

Rule: Defender

[Switch to Aggressor](#)

Each agent is the protector of a victim threatened by an aggressor. The agents move to position themselves between the victim and the aggressor.

Initially, each agent chooses, at random, a victim and an aggressor within its sight.


[Pick new partners](#) [Show/Hide relationships](#)

Population : 50

Sight : 22

Simulation speed : 58

[Restart](#)



Copyright © 2001, Icosystem. Reproduction is not permitted without the written consent of Icosystem.

The Bad News

- Cannot predict emergent behavior from individual rules, even for such a “simple” complex system
- Individual participants are unaware of overall system behavior
- Small changes in rules lead to dramatically different emergent behaviors

The Good News

- It is possible to manipulate the behavior of a complex system by changing the rules that control individual elements
- We have developed a methodology to predict emergent behavior in complex systems using bottom-up simulation

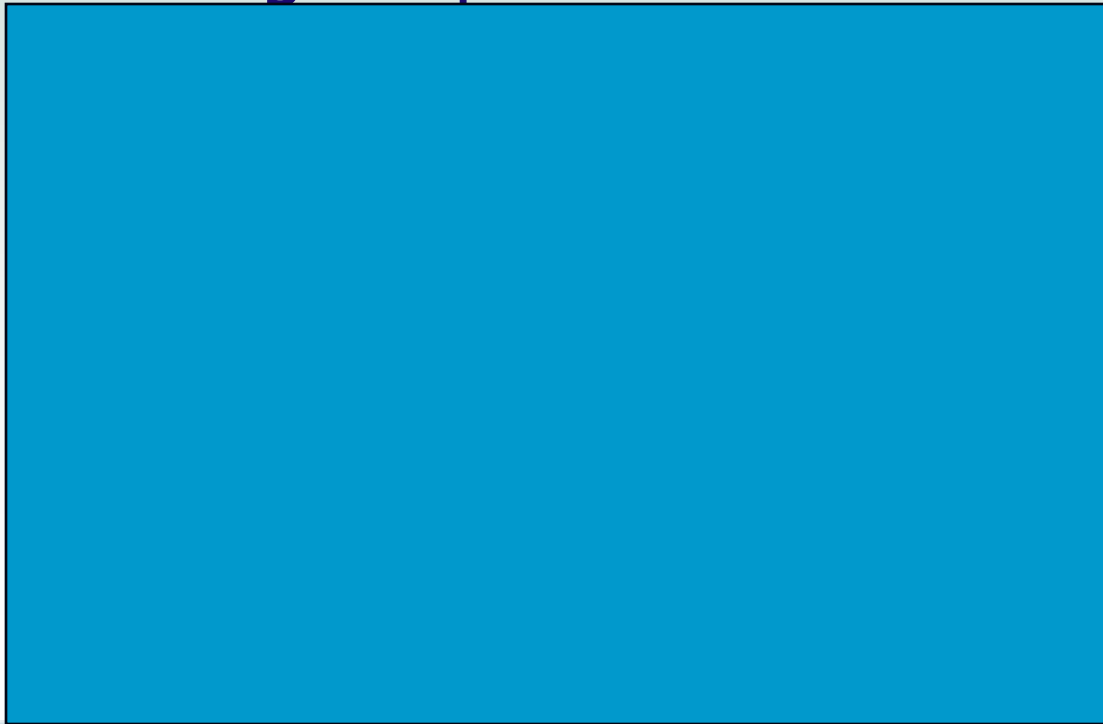
Agent-Based Modeling!

Sample Complex Systems



Controlling Emergent Behavior

- How can we control emergence?
- How do we define individual behaviors and interactions to produce desired emergent patterns?



“Here is
where we
think the
problem is...”

Agent-based modeling

- Shift viewpoint from *system* (centralized) to *individual elements* (decentralized)
- Each *agent* follows *local* rules
- Behavior depends on *interactions* with other agents
- Overall system behavior *emerges* from local interactions

Example: Flow Simulations

- Traditional approach: mathematical description at macroscopic level.
- Example: fire diffusion in airplane cabin



Limitations of Traditional Approaches

- Previous simulation requires extensive computation
- Any modification (e.g., number of seats, load, initial conditions) requires new computation

Compare to agent-based approach

Agent-based Flow Simulations

- The Game
- Boids
- Traffic



AGGRESSORS - DEFENDERS

Rule: Defender

[Switch to Aggressor](#)

Each agent is the protector of a victim threatened by an aggressor. The agents move to position themselves between the victim and the aggressor.

Initially, each agent chooses, at random, a victim and an aggressor within its sight.

[Pick new partners](#) [Show/Hide relationships](#)

Population : 50

Sight : 22

Simulation speed : 58

[Restart](#)

Copyright © 2001, Icosystem. Reproduction is not permitted without the written consent of Icosystem.

BIRDS

This is a simulation of bird flocking based on three simple local rules. Each boid is required to:

- 1) maintain a minimum distance from other objects, including boids,
- 2) match velocity with neighbouring boids and,
- 3) move toward the perceived center of mass of neighbouring boids.

Use the left mouse button to add obstacles.

[Restart](#) [Remove Obstacles](#)

Rule 1: Avoidance Weight : 100

Rule 2: Copy Weight : 60

Rule 3: Centroid Weight : 40

Population : 29

Simulation speed : 20

Swarm Control of UAVs

Supported by Air Force Research Labs SBIR

- Create Agent-Based Model of UAV swarm
- Test various swarm control strategies for two mission types:
 - Search (area coverage)
 - Search, track and hit targets (SEAD)
- Measure performance *systematically* under various scenarios and conditions

The UAV Agent-Based Model

- Rectangular search area
- 3-D motion: thrust, pitch, yaw control
- GPS for localization
- Probabilistic ground/target sensor
- Circular collision sensor
- *Pheromone* emitter & probabilistic sensor
- Communications (noisy) to central control
- Stationary or moving targets

Simulation: Area Coverage/Search

The screenshot displays the 'UAV Simulation' software interface. At the top, there are control buttons: 'Restart', 'Resume', 'One tick', and 'Shuffle targets'. The 'Simulation time' is shown as 574.0. Below these are several parameter settings, each with a slider and a shield icon: 'Seed: 0', 'Delta time (ms): 1000', 'Number of targets: 10', 'UAV max turning angle: 20', 'Sensor angle: 59', and 'Sensor radius: 100'. At the bottom left, two statistics are displayed: 'Proportion of ground covered: 0.1864' and 'Proportion of targets covered: 0.1'. On the right side, there is a 2D grid view showing a blue path and several red target markers. Below the grid is a 3-D perspective view of the simulation environment, showing a grey ground plane with red target markers and blue UAV icons. Three yellow callout boxes with arrows point to specific parts of the interface: 'Parameter Settings Panel' points to the parameter sliders, '3-D View Panel' points to the 3D perspective view, and 'Area Coverage Grid' points to the 2D grid view.

UAV Simulation

Restart Resume One tick Shuffle targets Simulation time: 574.0

Seed: 0

Delta time (ms): 1000

Number of targets: 10

UAV max turning angle: 20

Sensor angle: 59

Sensor radius: 100

Proportion of ground covered: 0.1864

Proportion of targets covered: 0.1

Parameter Settings Panel

3-D View Panel

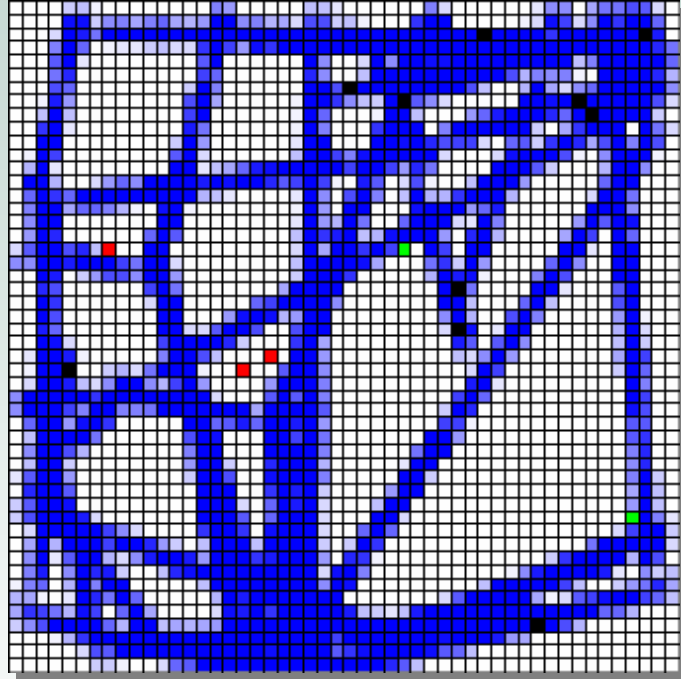
Area Coverage Grid

Navigation Strategies

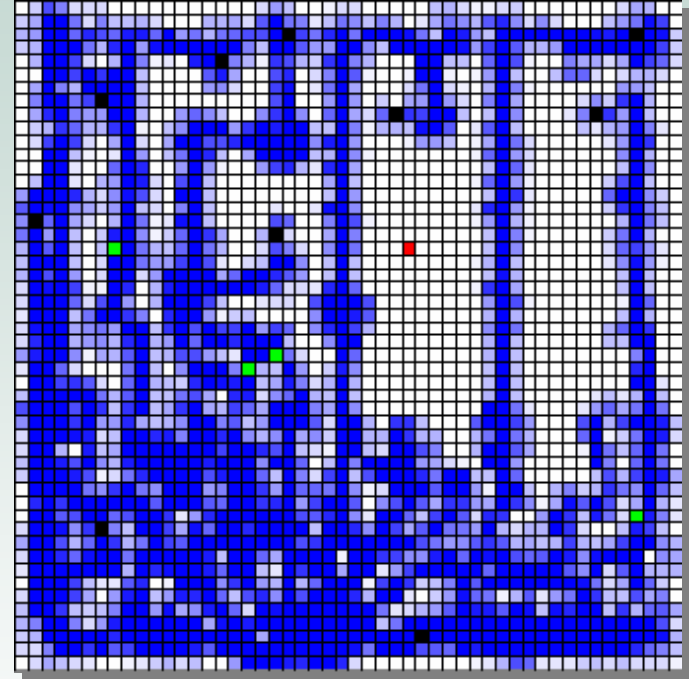
- *Baseline*: fly straight until border is detected, turn to stay within search area
- *Random*: inject small “jitter” in heading
- *Repulsion*: avoid UAVs within radius r
- *Pheromone*: avoid areas already covered (by self or others)
- *Global Search*: favor navigation toward unexplored sectors

(Strategies can be combined arbitrarily)

Sample Coverage Patterns



Repulsion ($r=60$)



Pheromone



Systematic Evaluation

Goal: Understand impact of strategies, parameter choices and scenarios:

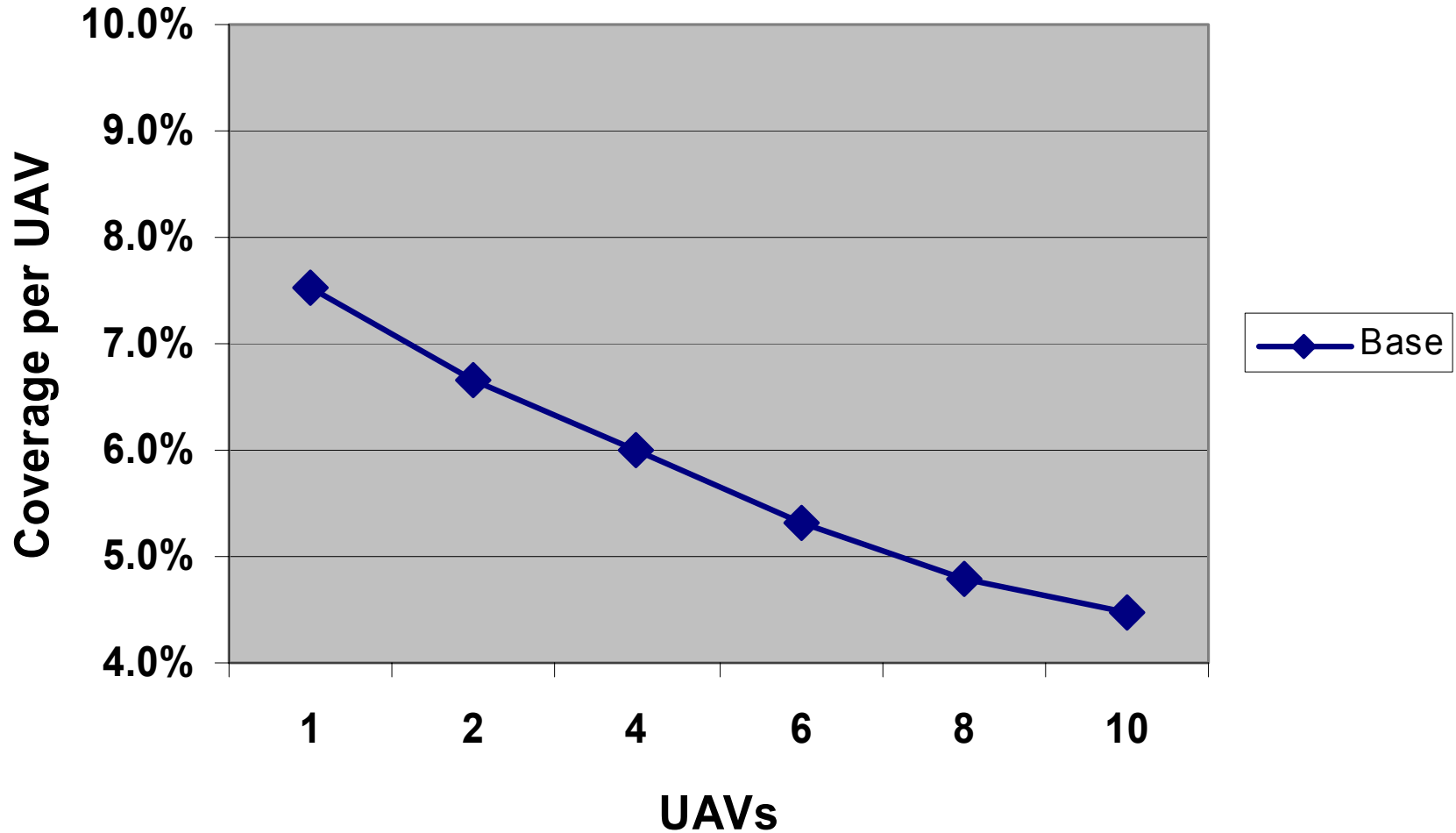
- 2000x2000 area, single UAV entry point
- 1000-sec simulation
- Swarm size (1-10, 10-110)
- Navigation strategies (individual & combo)

Metrics:

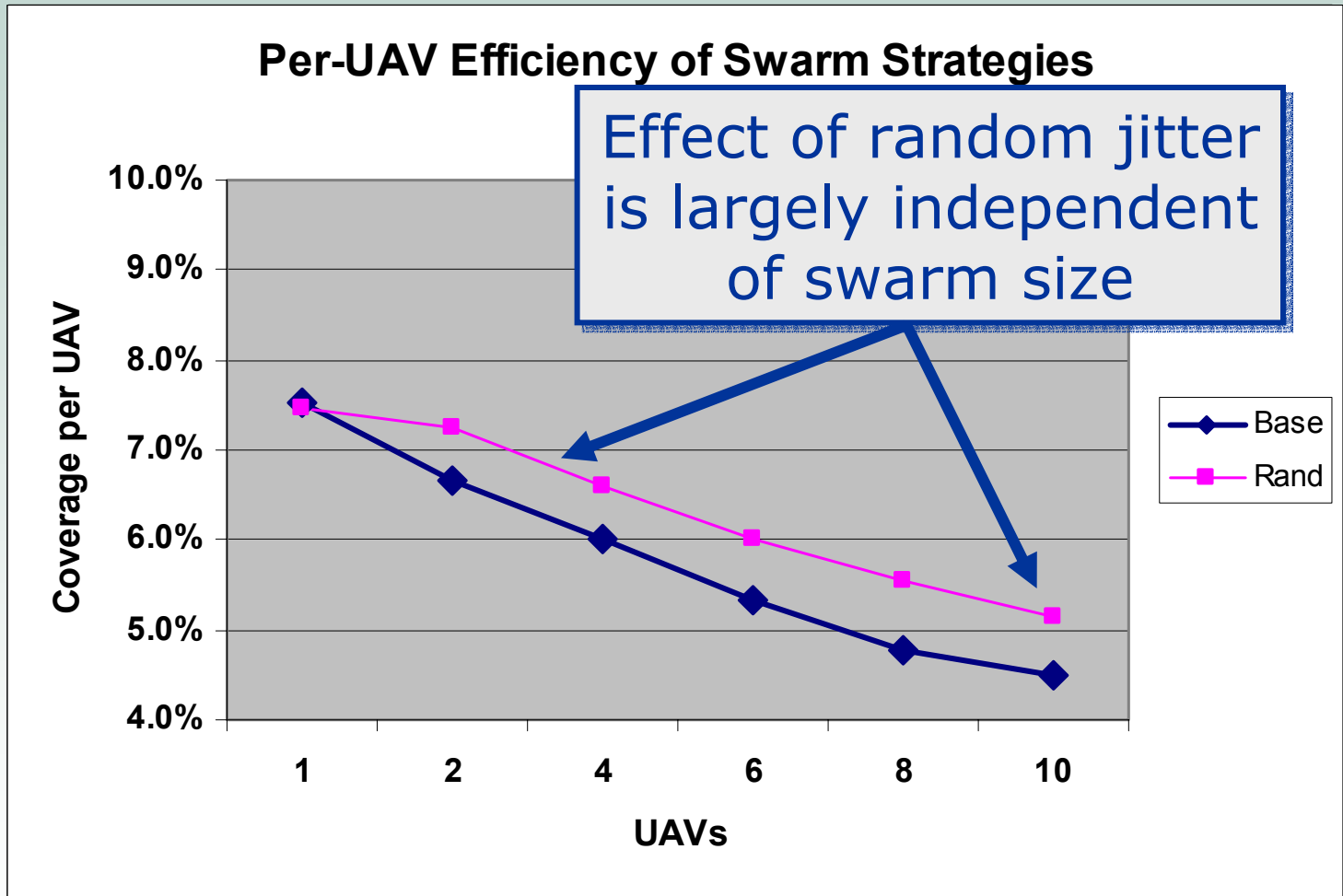
- Area coverage
- Swarm coverage efficiency
- Per-UAV coverage efficiency

Baseline Strategy

Per-UAV Efficiency of Swarm Strategies

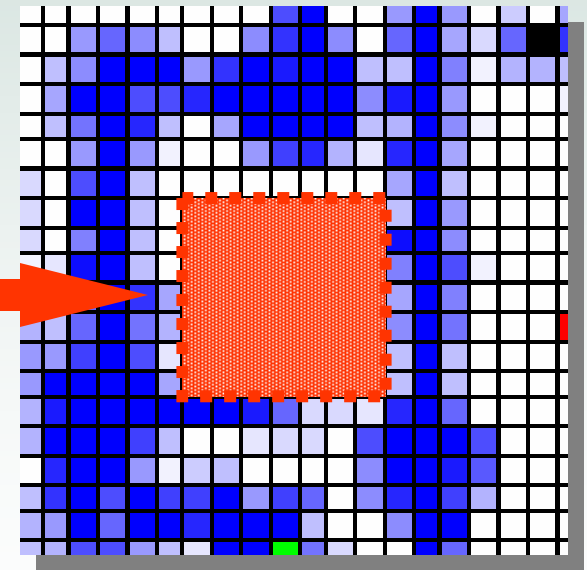


Random Noise Strategy

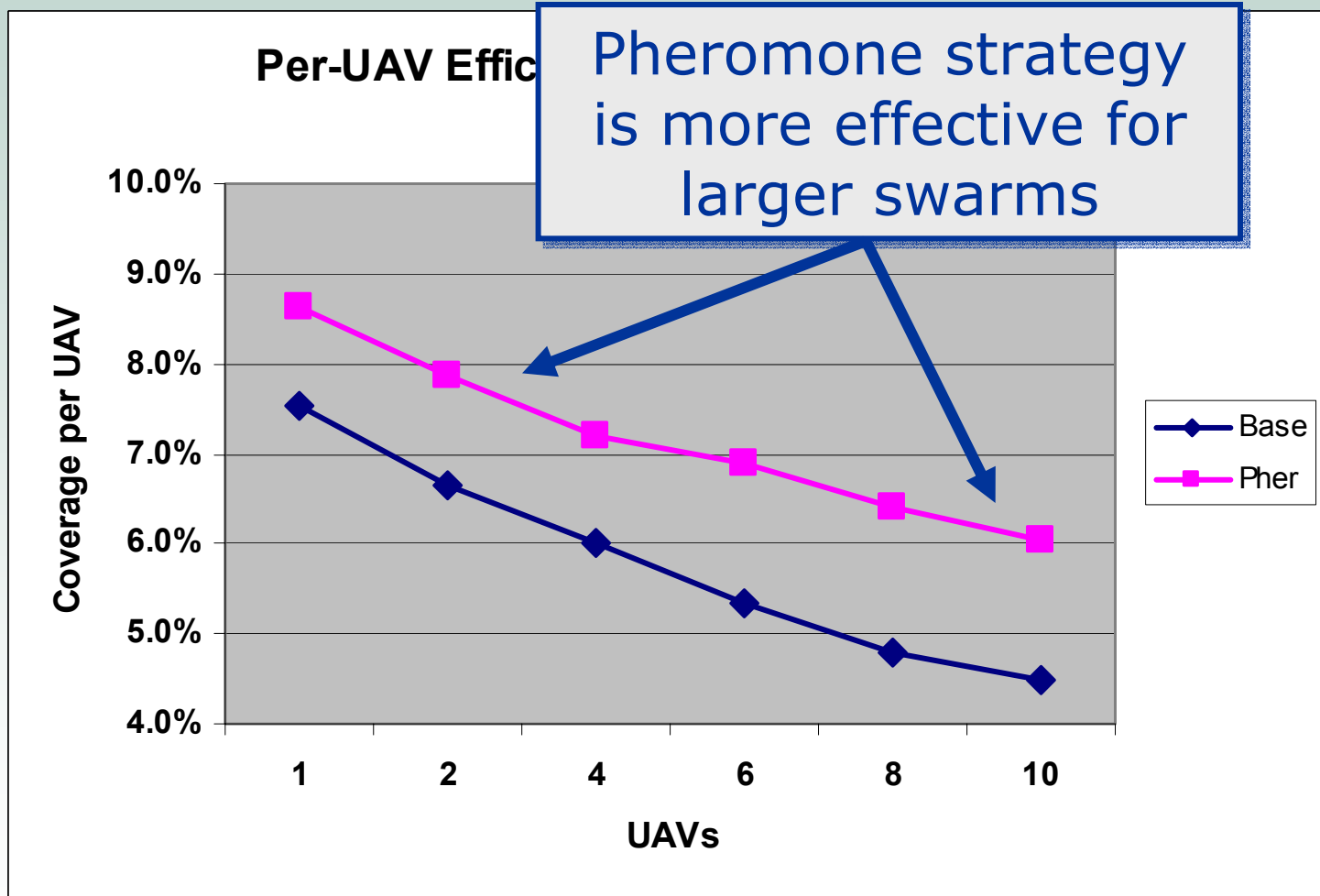


Pheromone Strategy

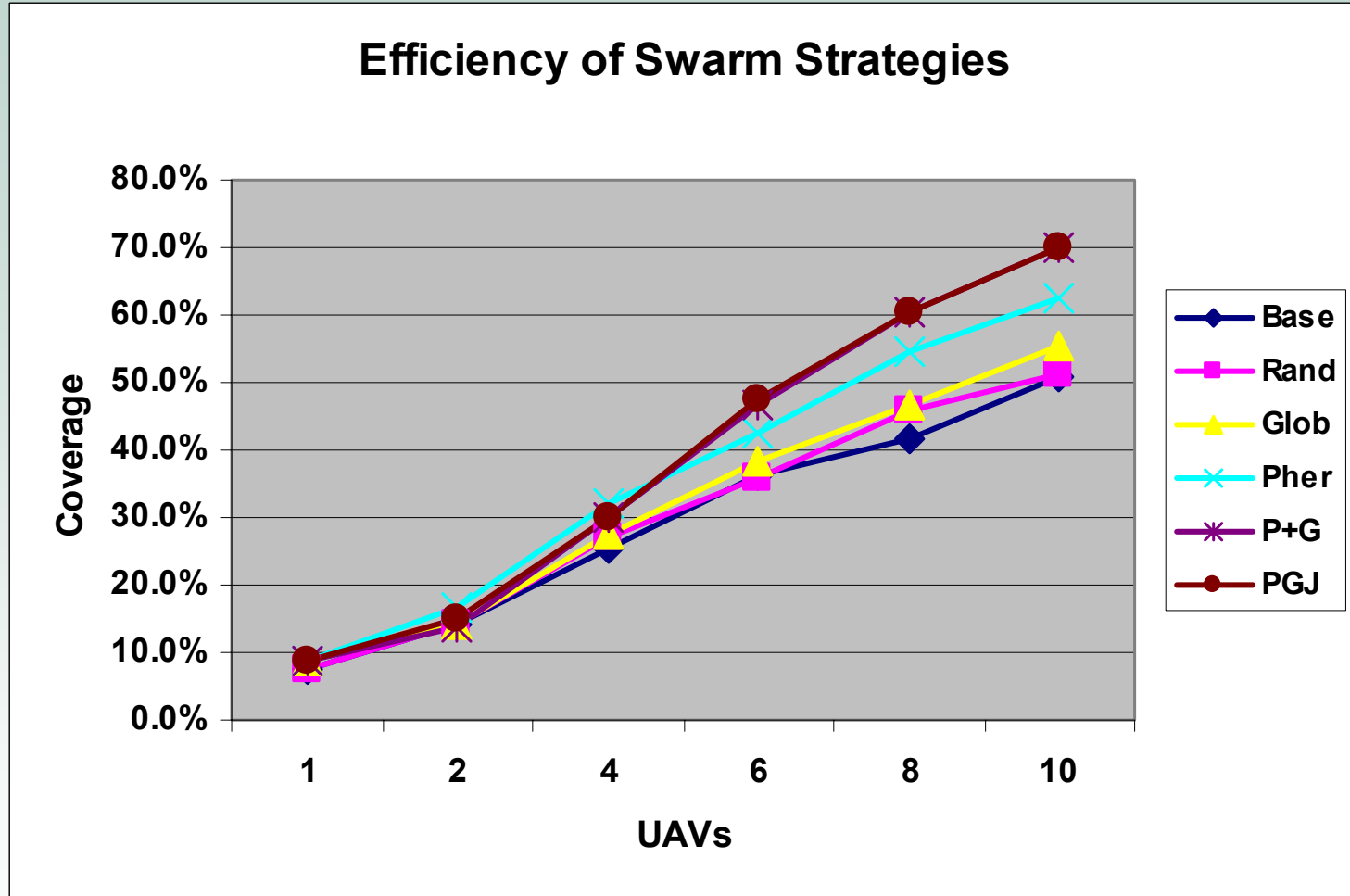
- Inspired by insect behavior
- Example of *stigmergy* (communication through the environment)
- Each UAV lays “pheromone”
- Each UAV can sense local pheromone trace
- Navigation favors uncovered areas (***Urea Strategy?***)



Pheromone Strategy Results

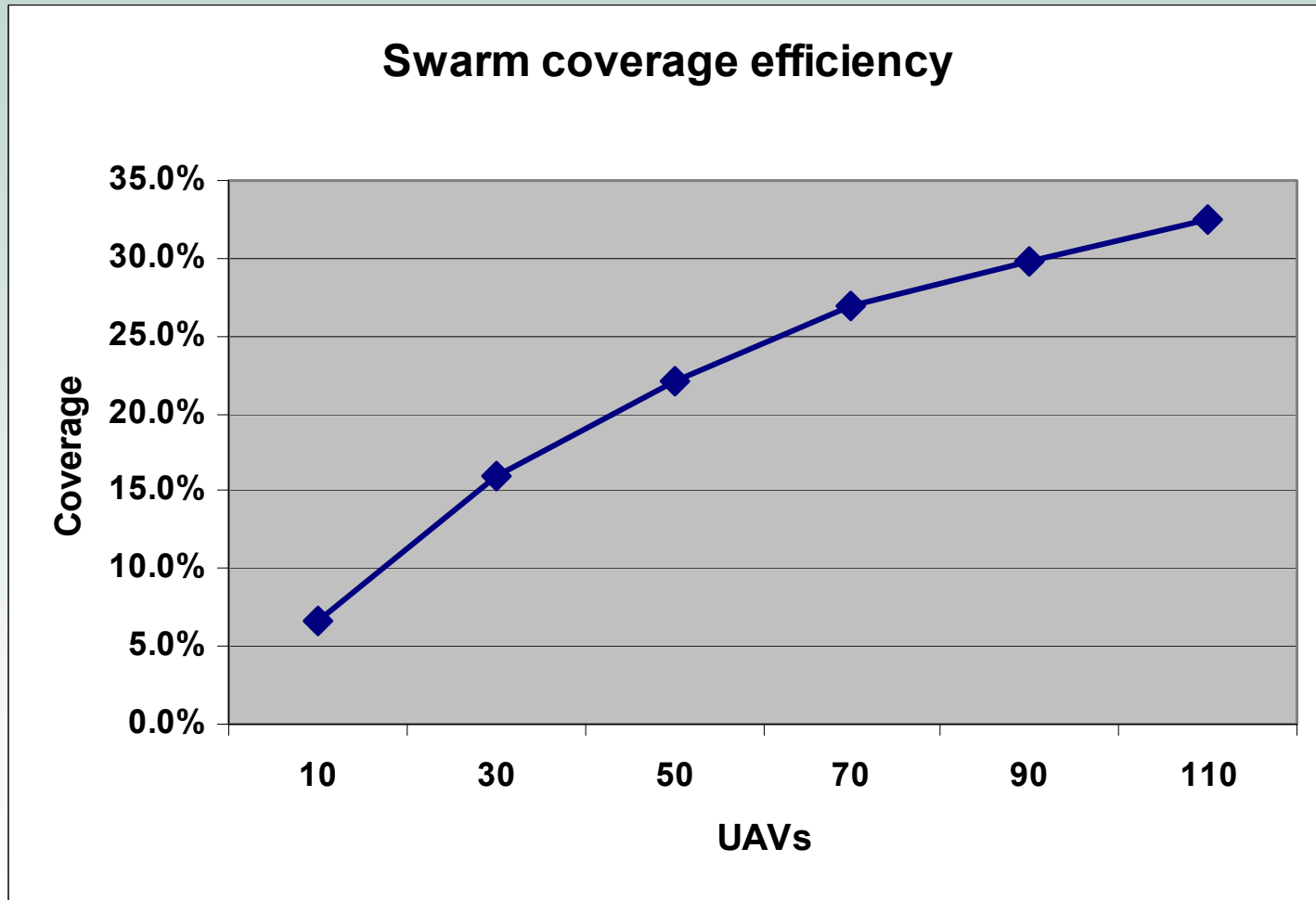


Combining Strategies



Even a relatively simple, *decentralized* strategy can yield significant improvement in swarm efficiency!

Extending to Large Swarms



Additional Results: SEAD

- Allow targets to move randomly over search area
- Extend UAV behavior to track targets
- Modify simulator to carry out *search and suppress* missions
- Apply *evolutionary computing* to identify robust strategies, parameters

Extended Simulator Demo

Unregistered Hyper Cam

Restart

Resume

One tick

Shuffle targets

Simulation time: 3.0

Seed: 0

ID under surveillance: -1

Delta time (ms): 1000

Number of targets: 12

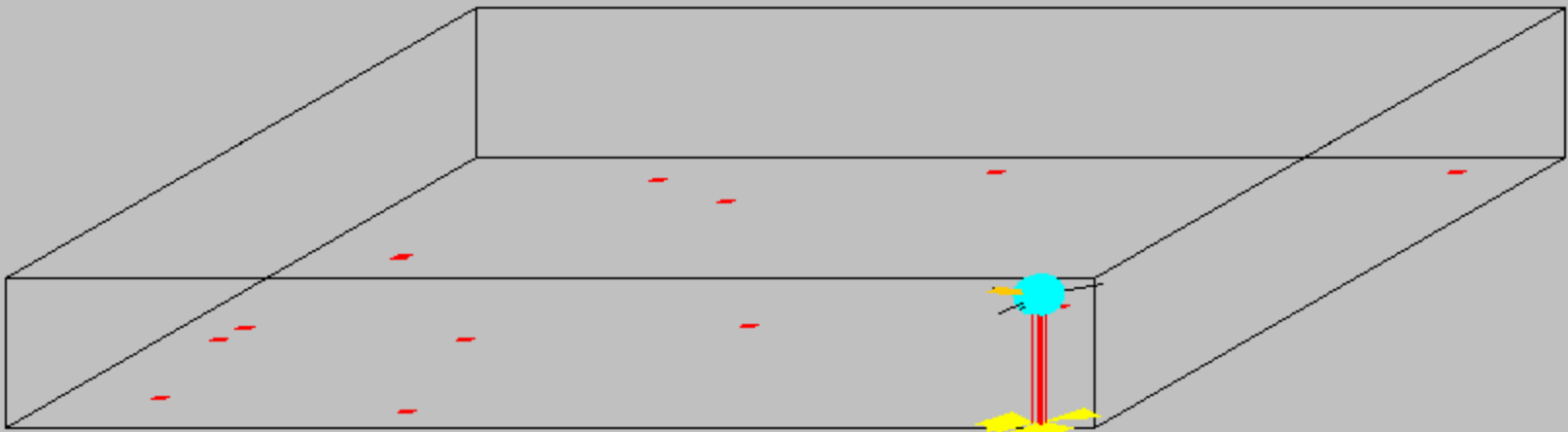
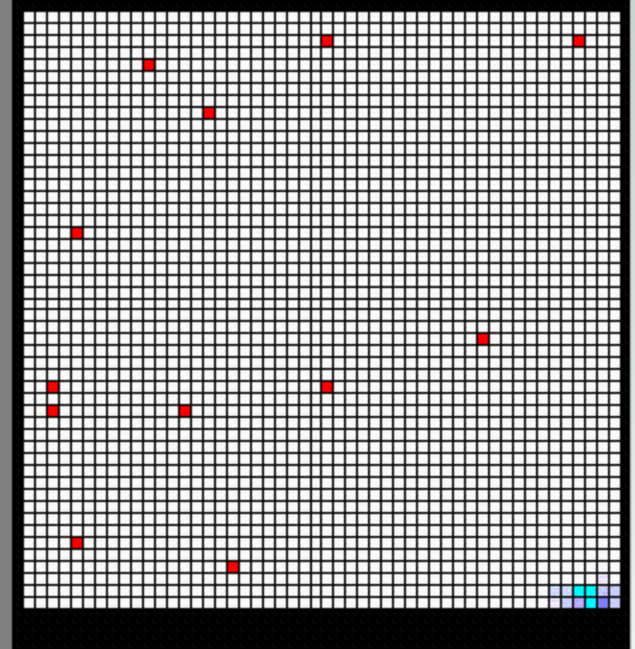
UAV max turning angle: 14

Sensor angle: 59

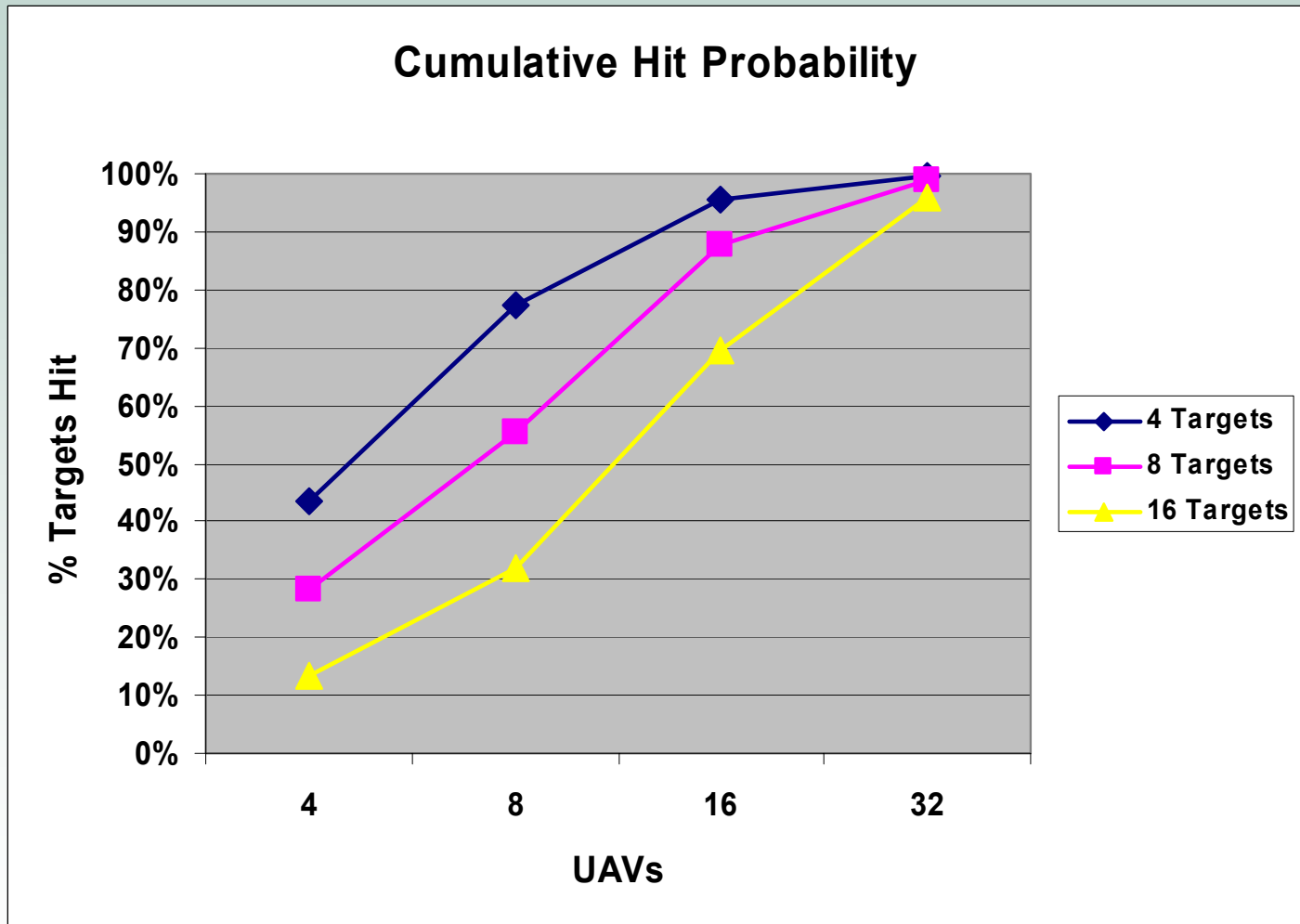
Sensor radius: 100

Proportion of ground covered: 0.0

Proportion of targets covered: 0.0



Sample SEAD Results



Future Work

- Systematic evaluation of other mission types, criteria, performance metrics
- Evolutionary design of control strategies
- Human-in-the-loop control
- Extend approach to *Unmanned Ground Vehicles* operating in urban scenario
- Commercialize these and other results