Cooperative Decision Making for Multirobot Systems

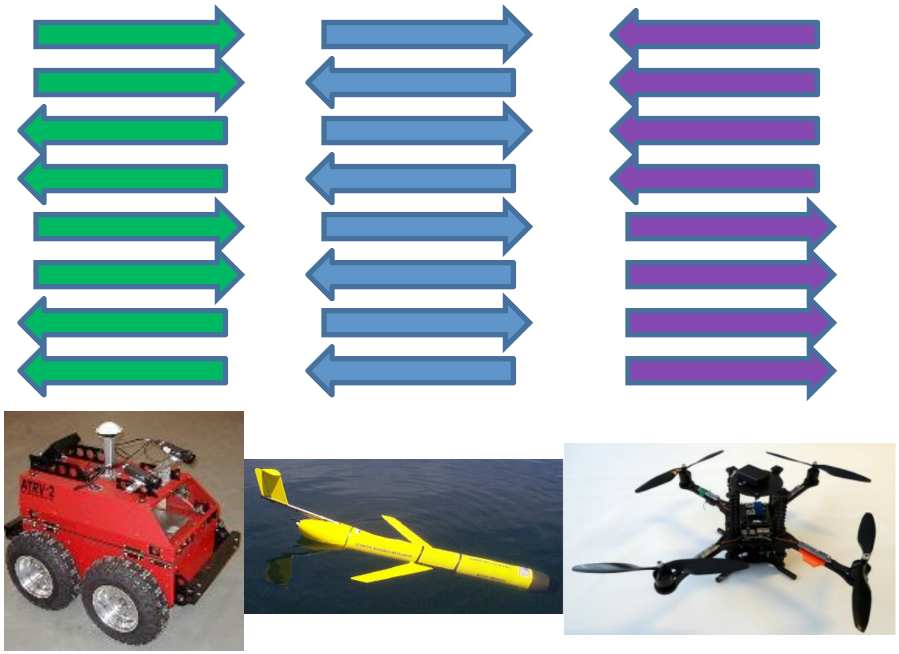
Geoffrey A. Hollinger

Assistant Professor
Robotics Program
Mechanical, Industrial & Manufacturing Engineering
Oregon State University

July, 2015

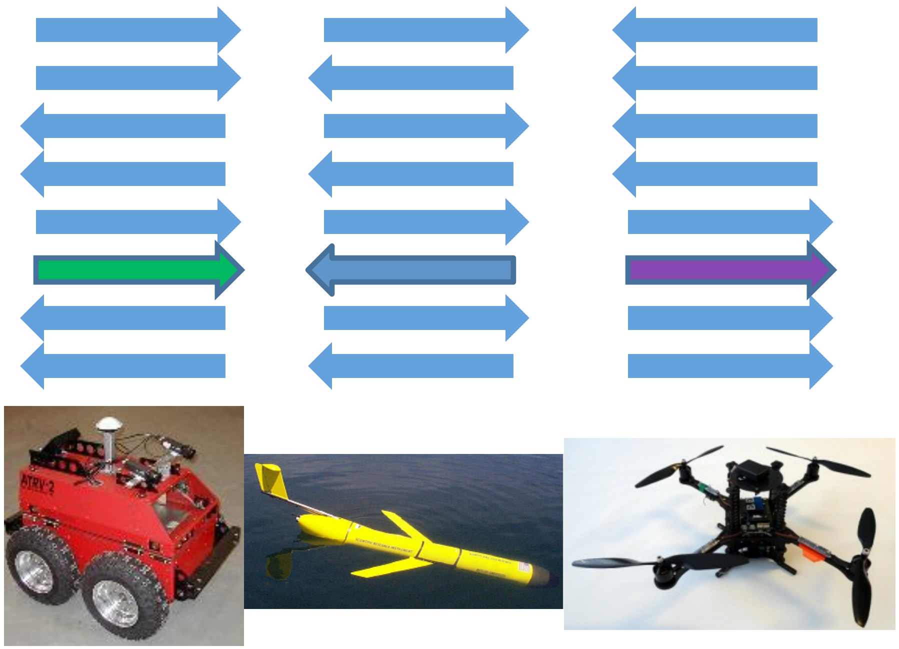
Why are multirobot problems hard?

- Choices increase drastically with more robots
  - Tasks require coordination between robots
  - Joint action-space grows with number of robots



Why are multirobot problems hard?

- Choices increase drastically with more robots
  - Tasks require coordination between robots
  - Joint action-space grows with number of robots



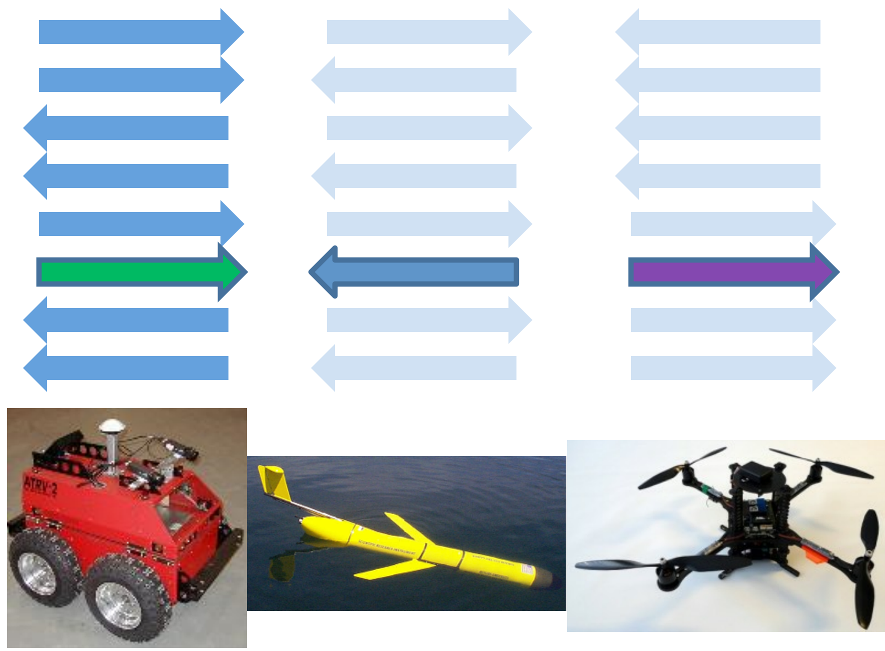
How can we solve multirobot problems?

Oregon State UNIVERSITY

- Centralized solver (Smith et al. ‘05, Kurniawati et al. ‘08)
  - One leader plans for the entire team
  - The leader tells everyone else what to do

Advantages

- Considers all possible strategies



Disadvantages

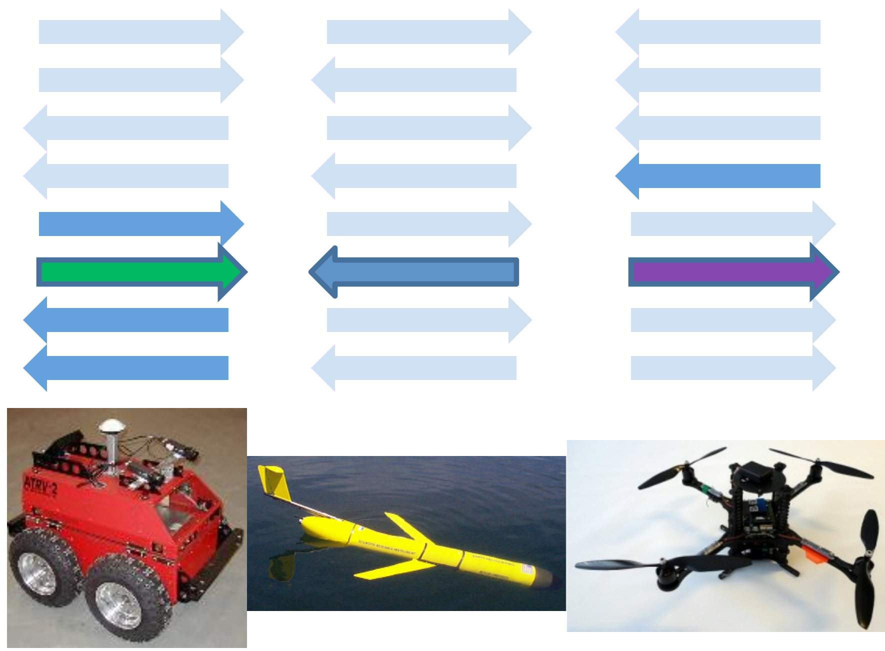
- High computation
- Single-point failure

How can we solve multirobot problems?

- Team coordination (Gerkey et al. ‘05)
  - Robots dynamically form and disband teams
  - Team leader plans for all robots on the team

Advantages

- Somewhat robust
- Relatively decentralized



Disadvantages

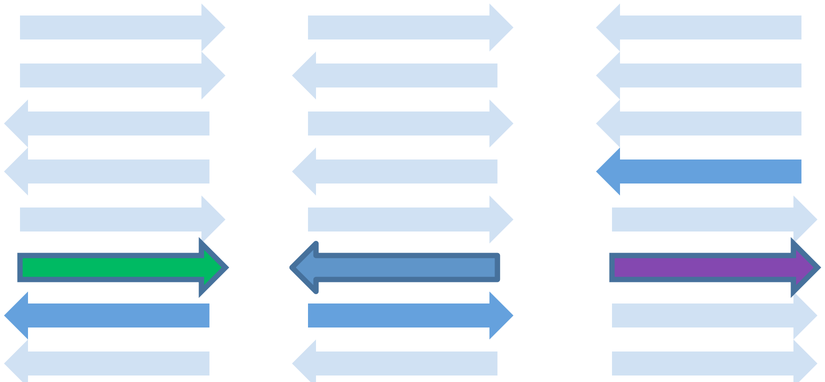
- Can still require high computation
- Need to determine when to form teams

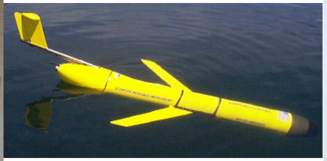
How can we solve multirobot problems?

- Implicit coordination (Grocholsky ’02, Hollinger et al. ‘08)
  - Each robot plans ***only*** its own actions
  - Robots communicate information to improve their actions
    - Examples: plans, measurements, target estimation

Advantages

- Scalable
- Robust
- Decentralized
- Little communication





Disadvantages

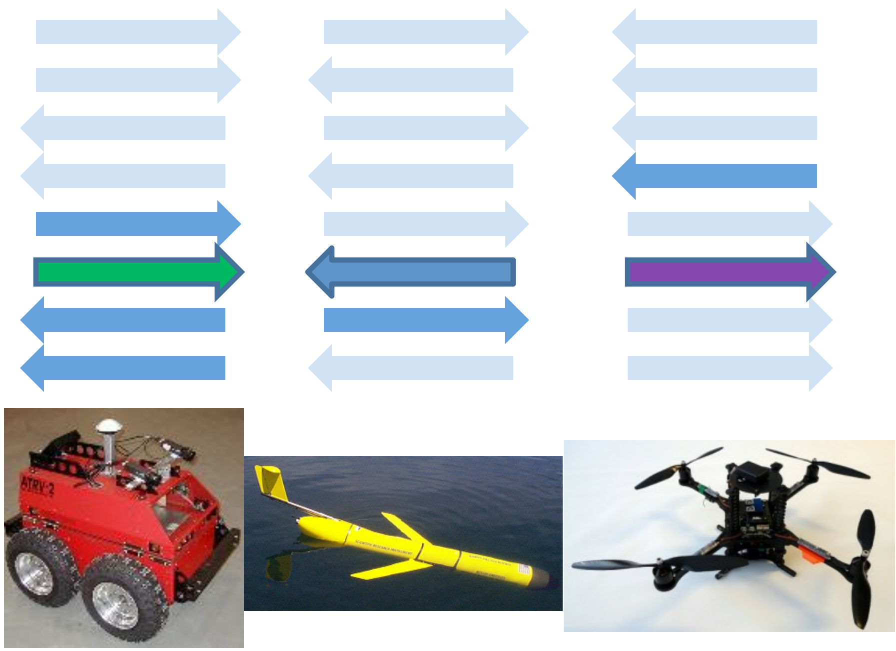
- May perform suboptimally

How can we solve multirobot problems?

- Market-based coordination (Kalra et al. ‘05, Zlot et al. ‘06)
  - Each robot plans for itself and for some teammates
  - Robots auction control actions to other robots

Advantages

- Relatively scalable
- Robust
- Decentralized



Disadvantages

- Higher communication
- Need to determine when to auction

Why decentralize?

- Centralized solvers
  - Single entity determines allocations
  - Requires reliable communication
- Decentralized solvers
  - Does not require central arbiter
  - Possible to generate ad hoc centralization for clique

Multirobot optimization

- **Problem statement**: given *K* robots with limited battery life B_k, efficiently perform a task in a bounded environment *W*
- **Assumptions:**
  - Bounded planar workspace *W*
  - Workspace is divided into free regions W_{free} and obstacle regions W_{obs}
  - Workspace partition into obstacle and free could be initially unknown
  - The robots are equipped with a sensor that allows them to observe the environment with limited range and visibility radius

Optimization Representation

- Multirobot coordination can be cast as an optimization of the form

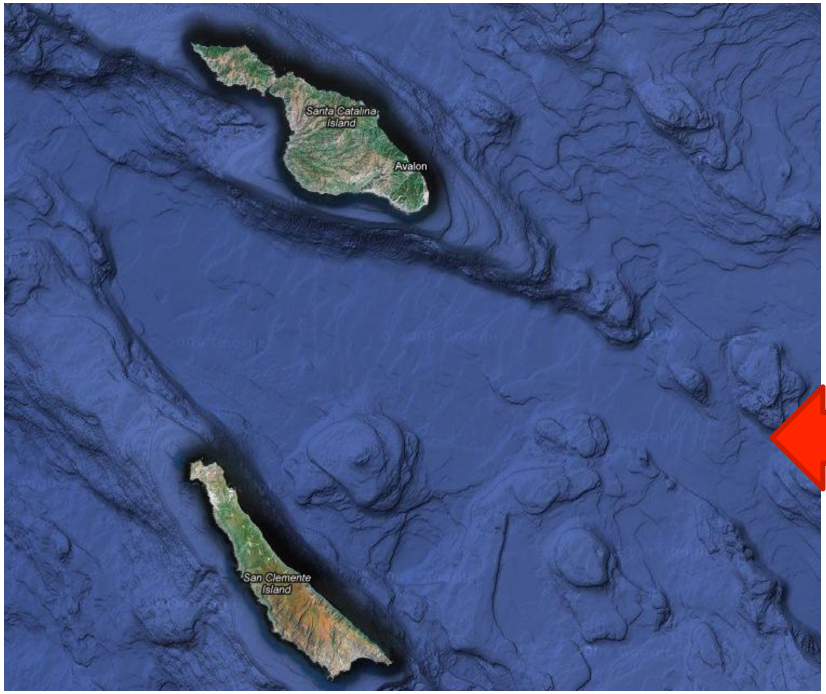
$$P^* = argmax_P \; R(P) \; s.t. \; |P_k| < B_k \; for \; all \; k,$$

- *R(P)* is a reward metric related to the task completed by a set of trajectories *P*
- $|P_k|$ is the battery consumed by a trajectory P_k for robot *k*

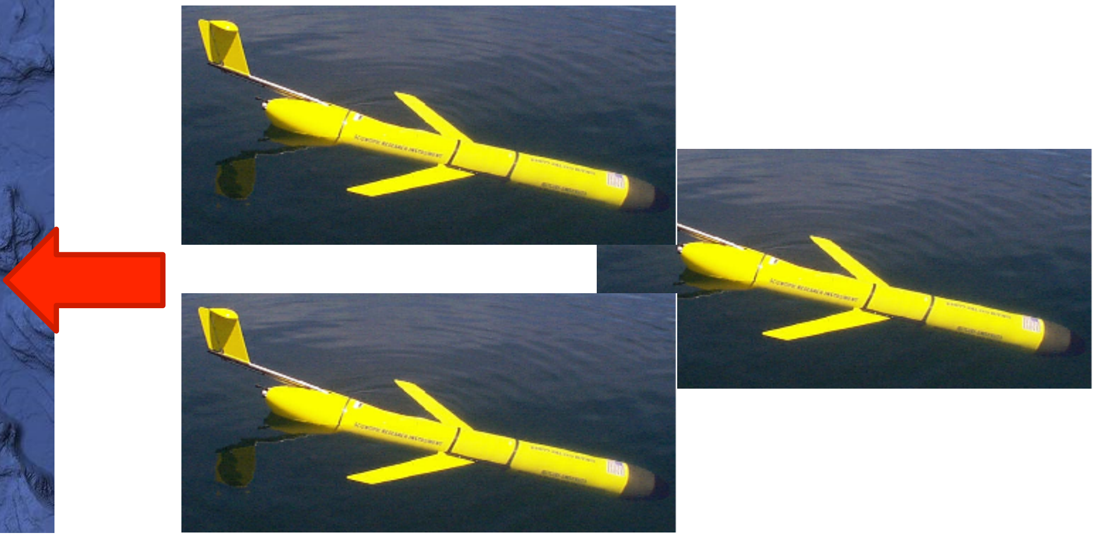
Reward Metrics

- **Search**
  - Probabilistic: maximize probability of capturing one or more targets
  - Guaranteed: search environment such that worst-case target could not escape
- **Exploration**
  - Observe as much of the environment as possible
- **Mapping**
  - Maximize map accuracy in limited time
    - Average accuracy
    - Worst-case accuracy
- **Note:** objectives may not be aligned (i.e., maximizing one may not maximize another)

Example domain: Multi-robot search



Where are the targets of interest?
Are there interesting events occurring?



- Multiple vehicles gather information about the environment
  - Sharing information during the mission can improve performance (e.g., state, observations, plans)
  - In many cases communication is limited by configuration

Modeling the search problem

- Discounted reward metric: $F_Y(A) = \gamma^t$

Target path

Searcher paths

Discount factor

Capture time

Modeling the search problem

- Discounted reward metric: $F_Y(A) = \gamma^t$
- Probabilistic motion model (average-case)

$$\operatorname*{argmax}_A \sum_{Y \in \Psi} \Pr(Y) F_Y(A)$$

- Adversarial model (worst-case)

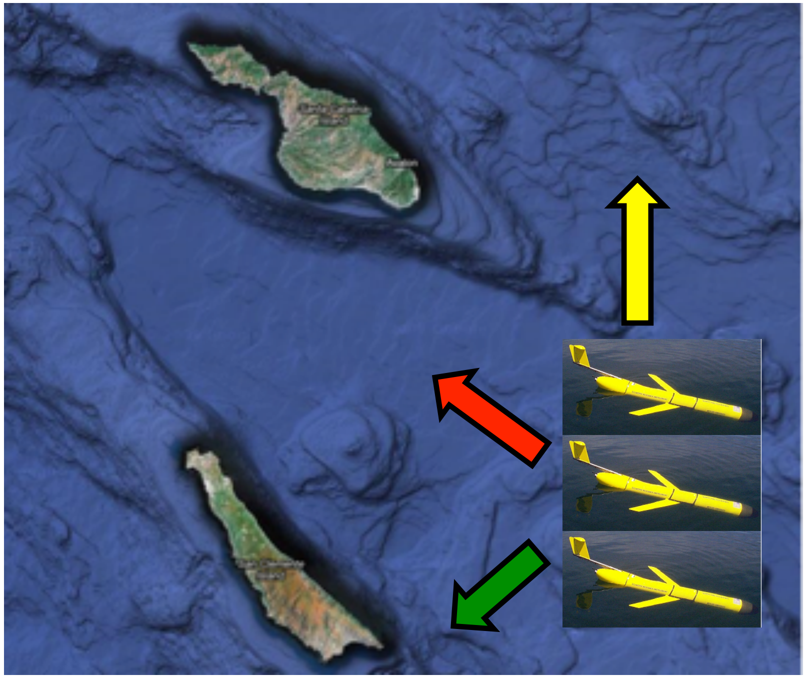
$$\operatorname*{argmax}_A \min_Y F_Y(A)$$

Search optimization

- Maximize discounted probability of capture

$$A^* = \underset{A}{\operatorname{argmax}} \sum_{Y \in \Psi} \Pr(Y) F_Y(A) \text{ where } F_Y(A) = \gamma^t$$

- Receding-horizon planning
  - Examine paths within fixed depth
  - Replan
- Implicit coordination
  - For all robots
    - Plan and share path with others
    - Others assume shared paths fixed
- Linear scalability in team size



(Hollinger et al. IJRR '09)

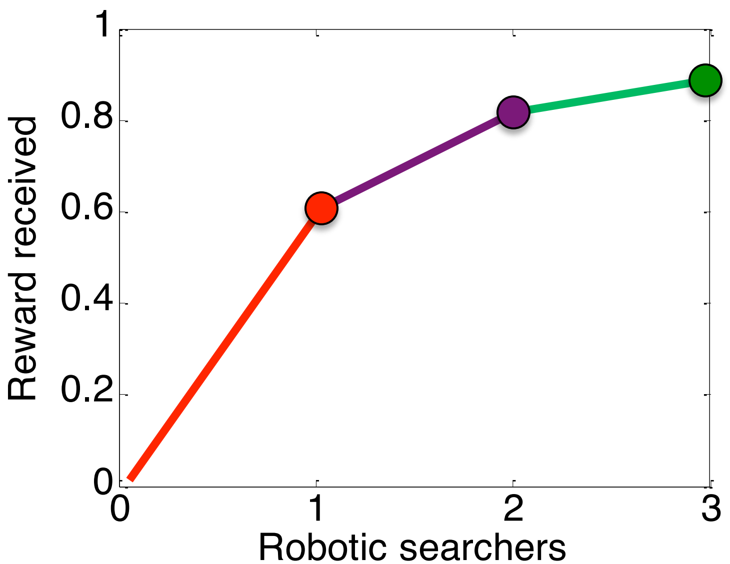
Performance guarantees

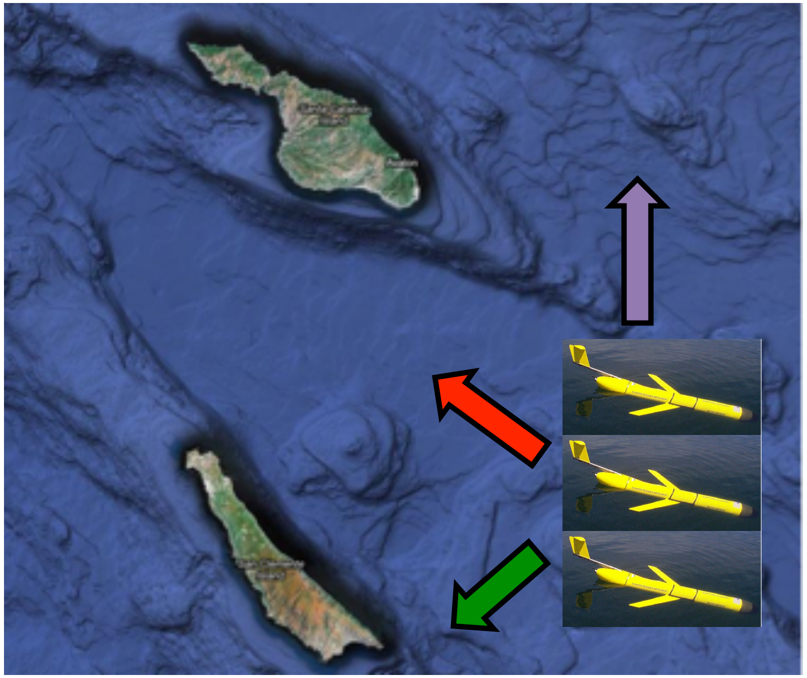
- Given a set function (e.g. discounted capture probability)
- **Submodular** if:

$$A \subseteq B \Rightarrow F(A \cup C) - F(A) \geq F(B \cup C) - F(B)$$

- **Nondecreasing** if:

$$A \subseteq B \Rightarrow F(A) \leq F(B)$$

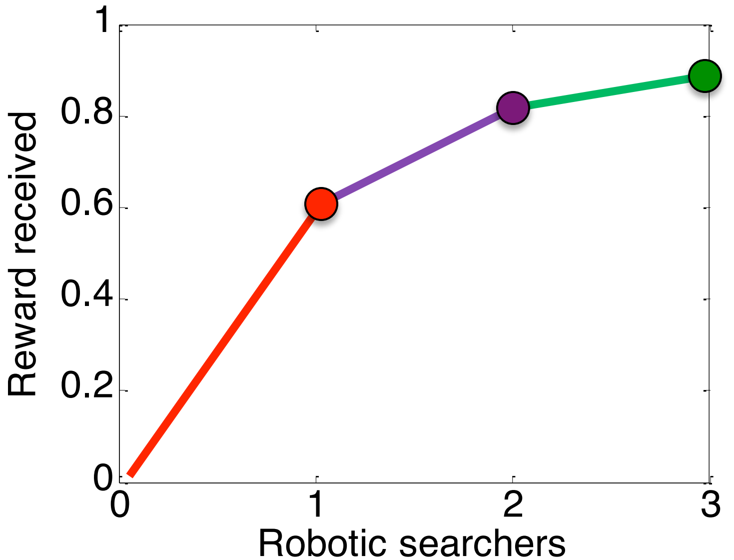


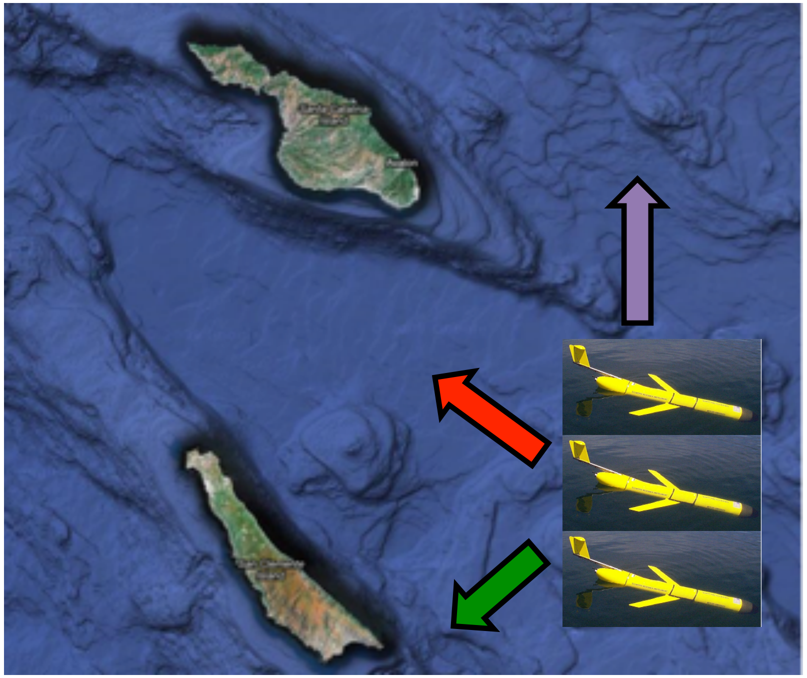


Performance guarantees

- **Theorem**: Average-case search optimizes a **nondecreasing**, **submodular** set function
- This means: Implicit coordination yields a performance guarantee (over the horizon)

$$F(A^{SA}) \geq \frac{1}{2} F(A^*)$$

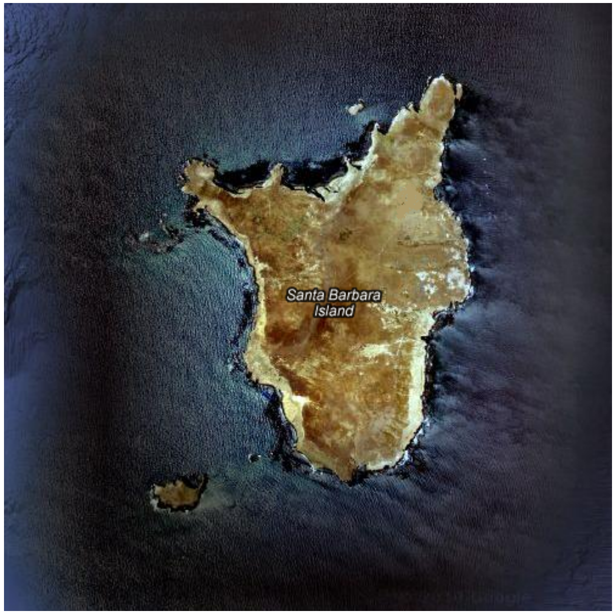




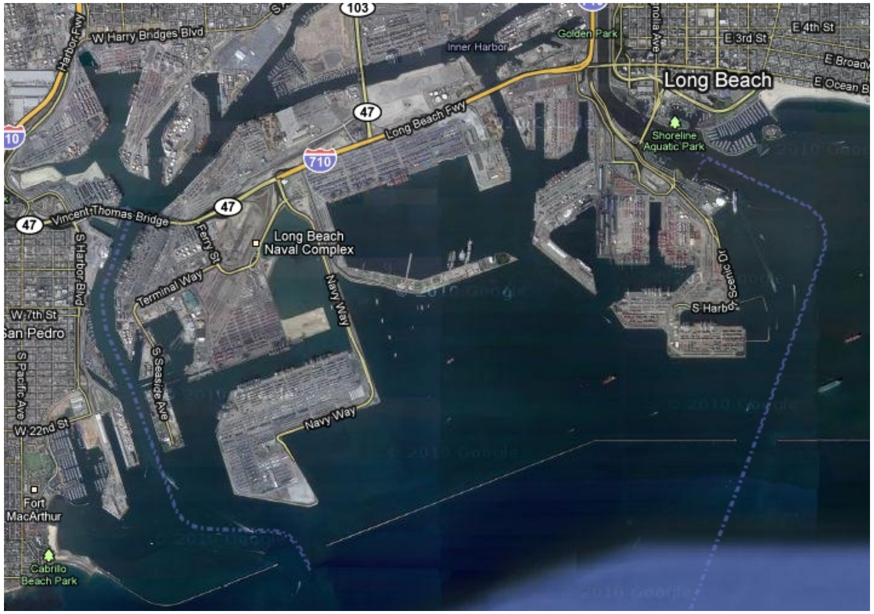
(Hollinger et al. IJRR '09)

Underwater search simulations

- Simulated island and harbor environments
- AUVs must locate a target at a fixed depth
- Each robot plans using its current belief
  - Communication through acoustic channel
  - Allows for “opportunistic” belief merging



Santa Barbara island: 4 km x 4 km

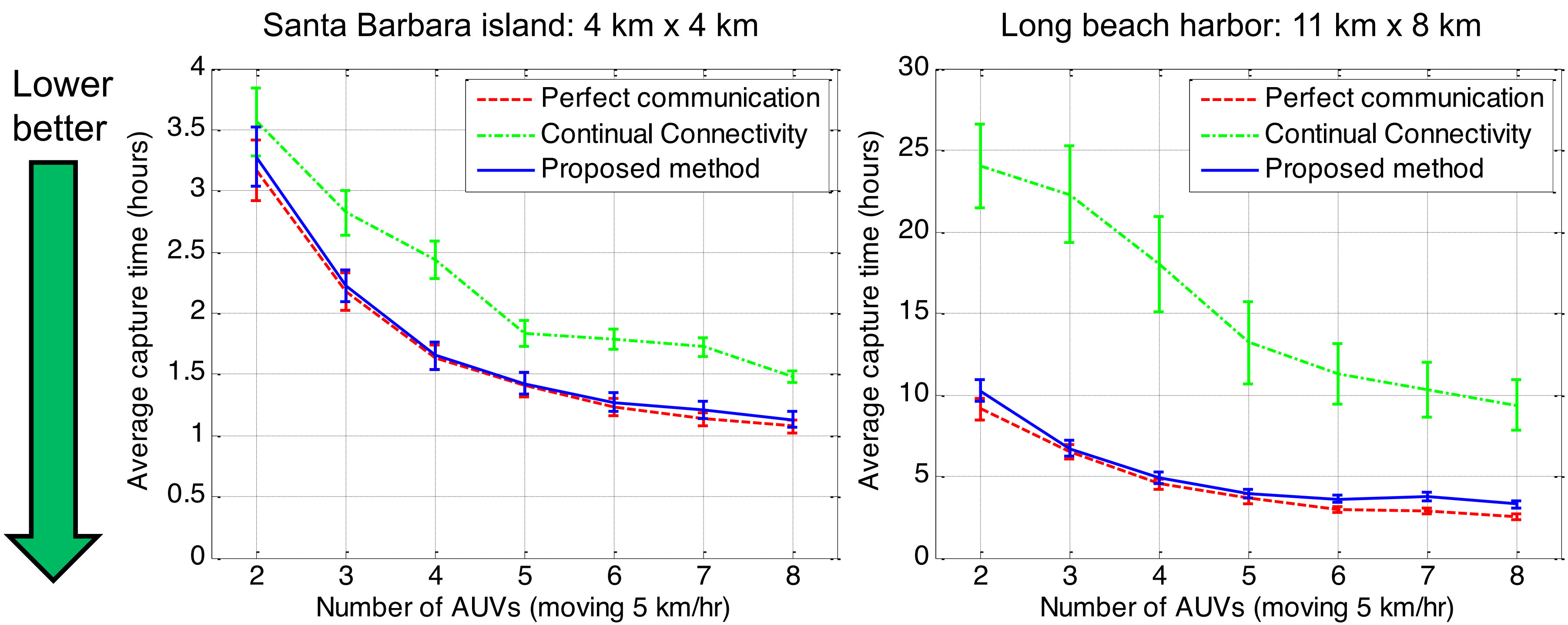


Long beach harbor: 11 km x 8 km

Underwater search video

Underwater search results

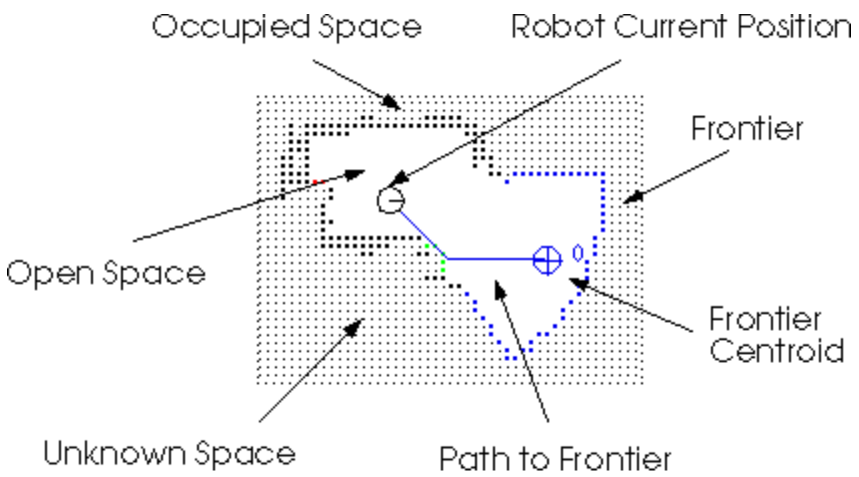
- Perfect communication: unrealistic baseline
- Continual connectivity: conservative planning for full communication
- Proposed method: implicit coordination with data fusion



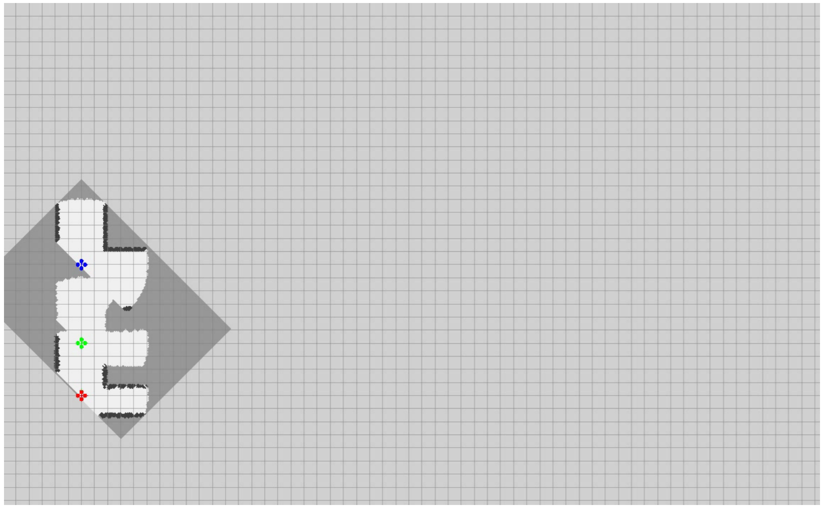
Each data point averaged over 200 simulations

Example domain: exploration

- Finds open cells next to unknown cells
  (Yamauchi ’97, Burgard et al. ’00)
- Uses blob detection to identify frontier regions
- Assigns robot to explore nearest frontier region
  - Alternatives: TSP, market-based, etc.

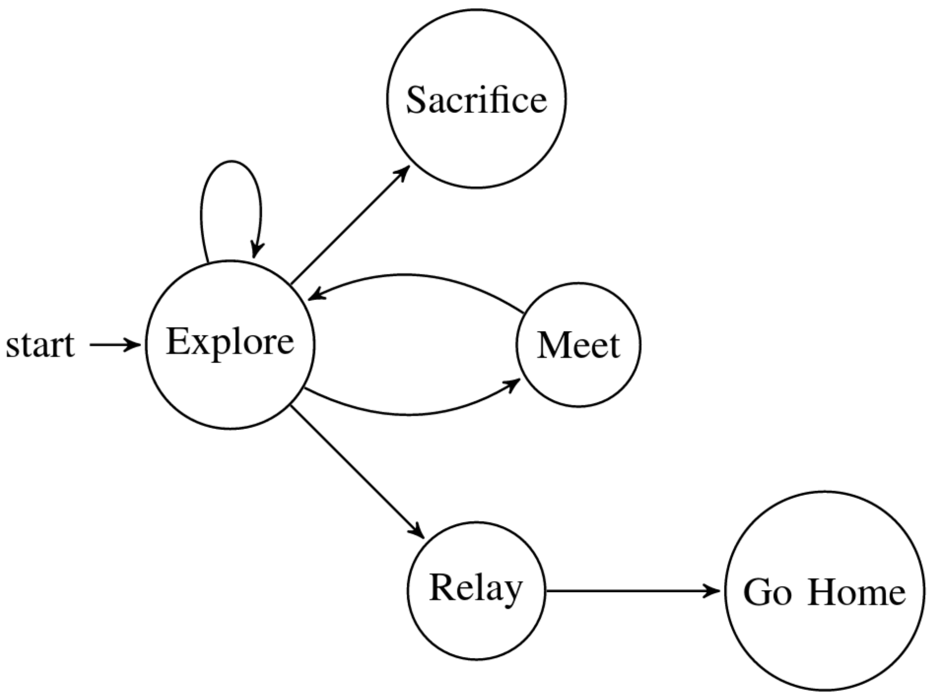


*Taken from robotfrontier.com



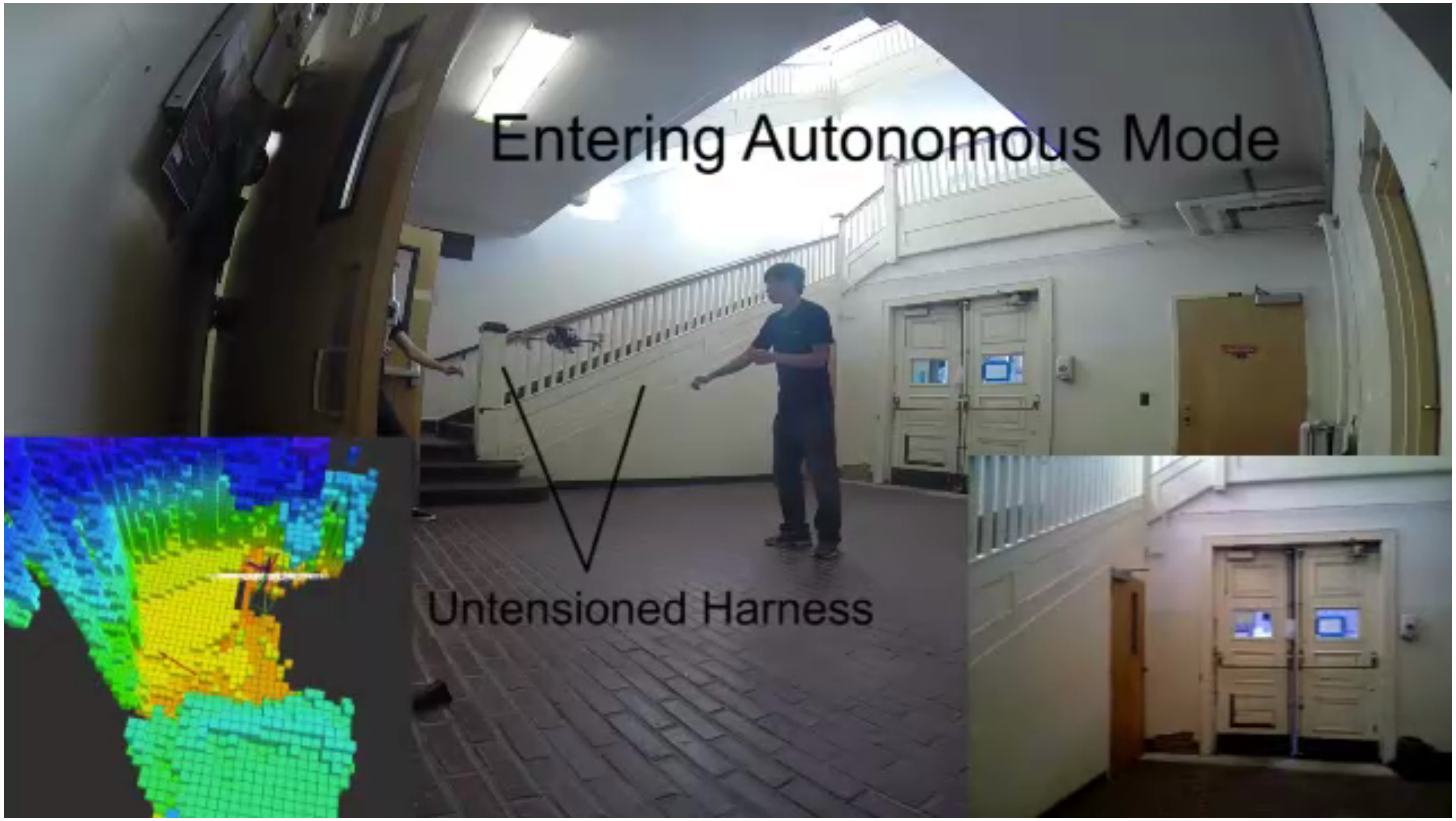
State machine

- Coordinates robots
- Robust to unreliable communication
- Considers limited battery life
- Add additional states for “sacrifice” and “relay” capabilities (Cesare et al. ’15)



Autonomous quadcopter mapping

- Two custom quadcopter UAVs mapping a building



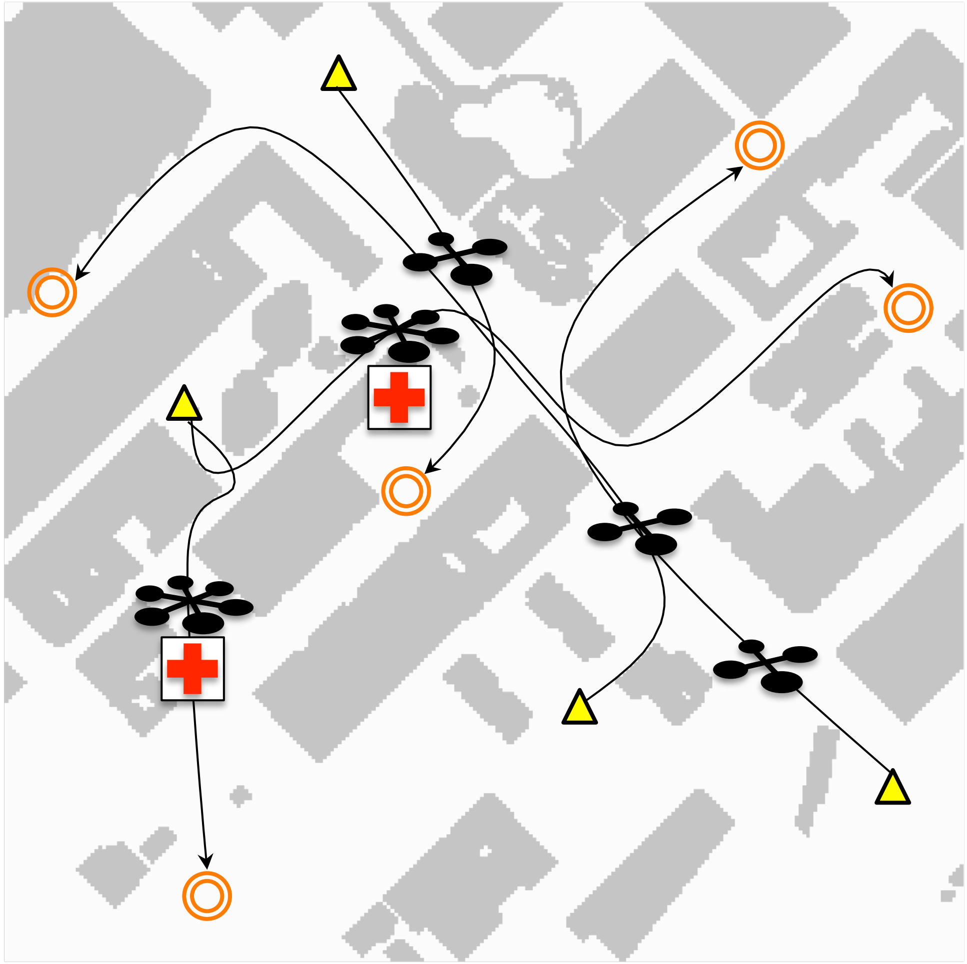
(Cesare et al., ICRA 2015)

Example domain: UAV package delivery

- Increasing popularity of delivery drones: UPS, Amazon, etc.
- Dense UAV traffic in cluttered urban environment
- No current framework for large scale coordination

A Cross-Section of the Airspace

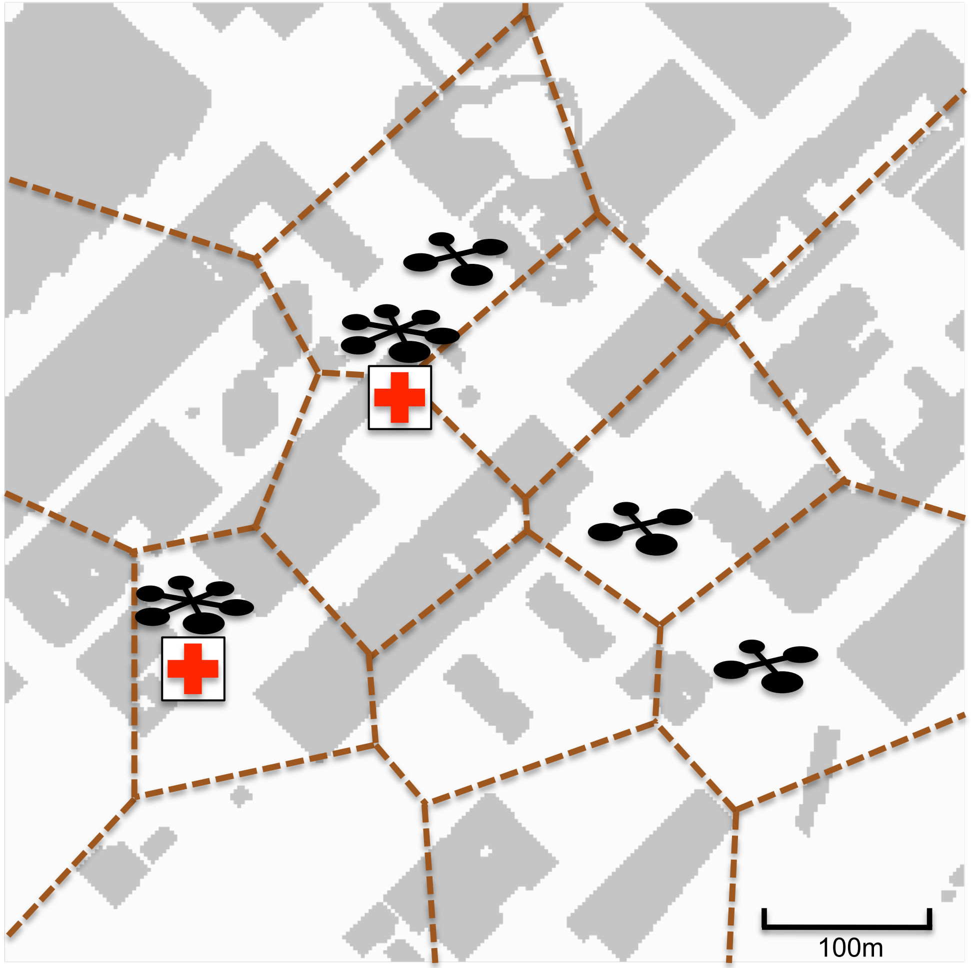
- Automated UAV traffic management
- Challenges:
  - Narrow thoroughfares of dense traffic
  - Heterogeneous UAVs
  - Dynamic obstacle landscape
- Goals
  - Minimize conflict occurrences
  - Avoid cascading effects
  - Maintain throughput



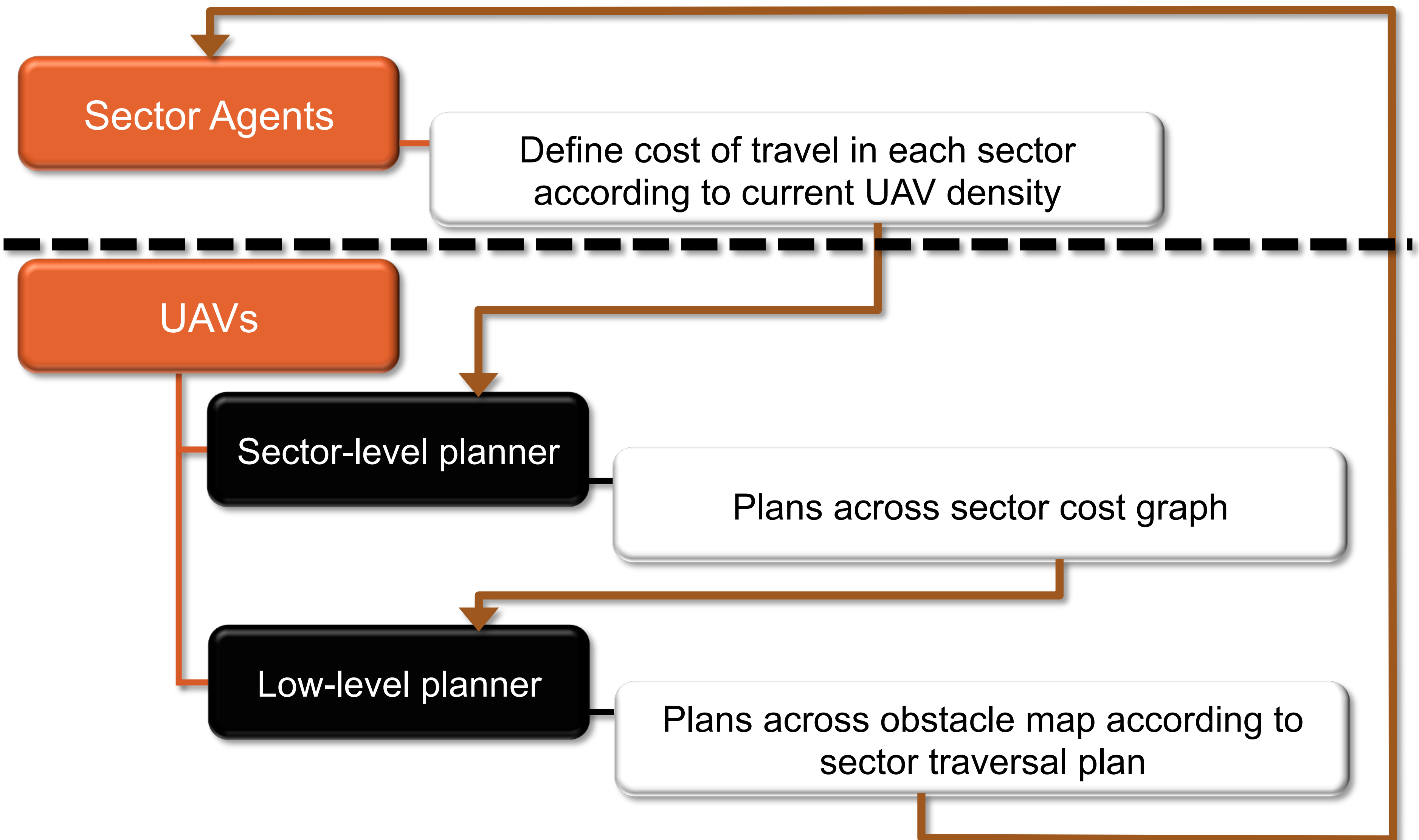
Multiagent UAV Traffic Management (UTM)

- Divide airspace into sectors
  - Assign single agent to manage each sector

- Multiagent team:
  - Agents **individually** learn policy for assigning sector traversal costs
  - Reward is total number of conflicts in **global** system



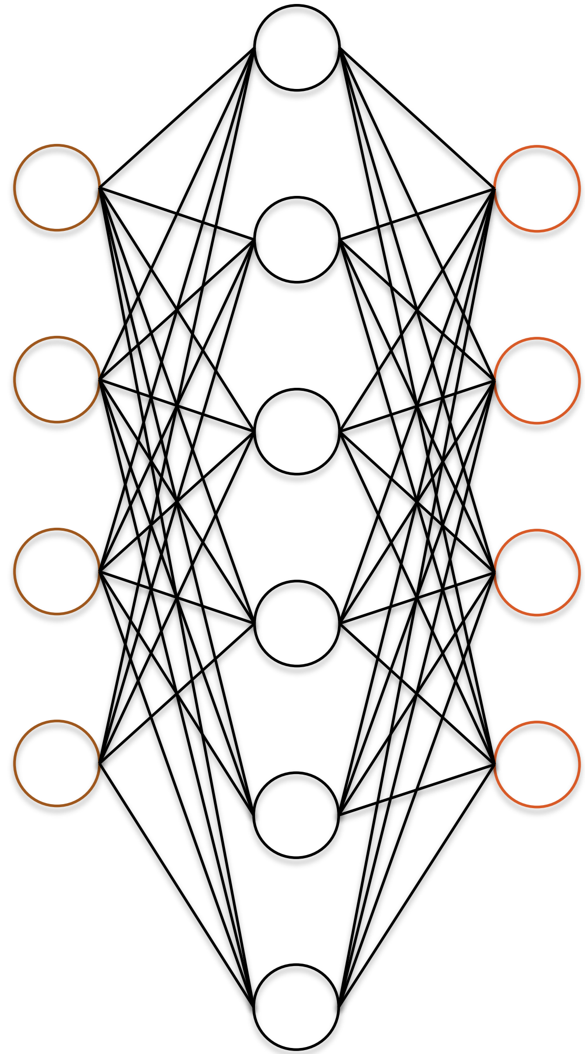
A Hierarchical Approach



UTM Learning Agents

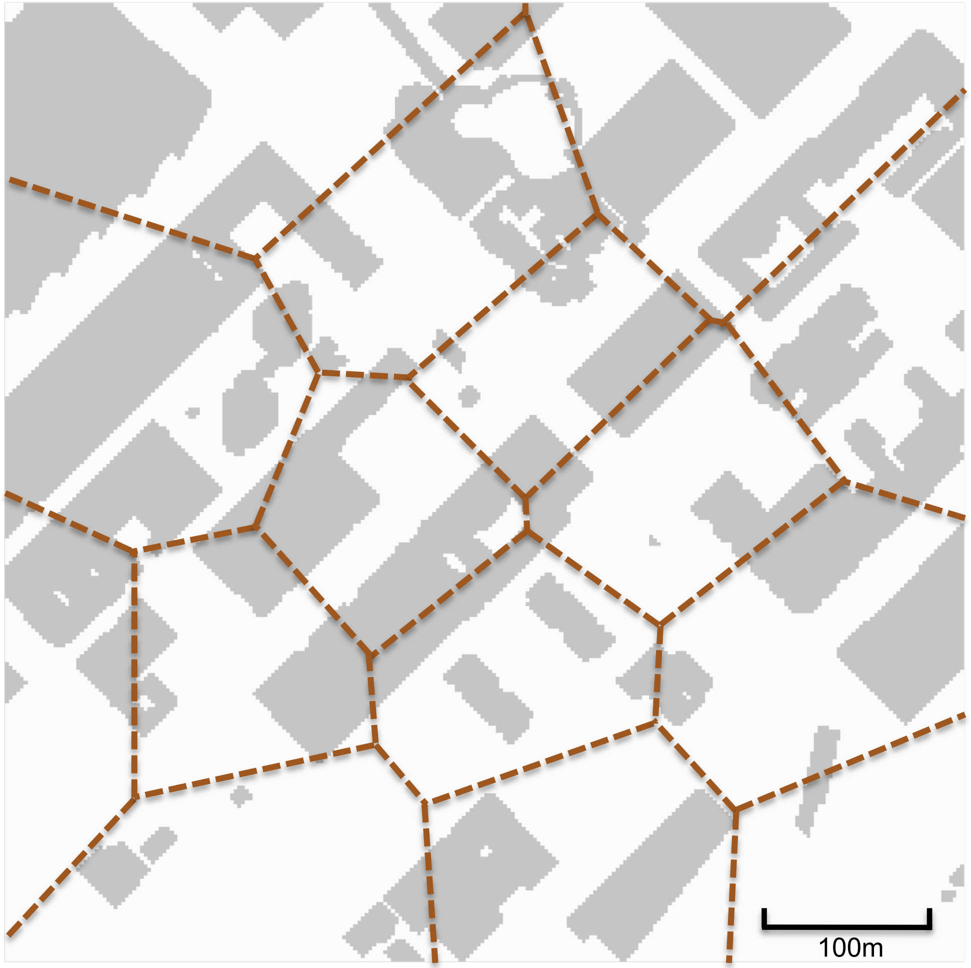
Oregon State UNIVERSITY

- Learn the cost of travel to apply to UAVs in the sector
- Neural network control
  - Inputs: UAV counts in sector
    - Separate into traffic types, e.g. heading, priority, platform etc.
  - Outputs: Cost of through-sector travel for each traffic type
- Cooperative coevolution to learn NN weights
  - Fitness value: number of conflicts

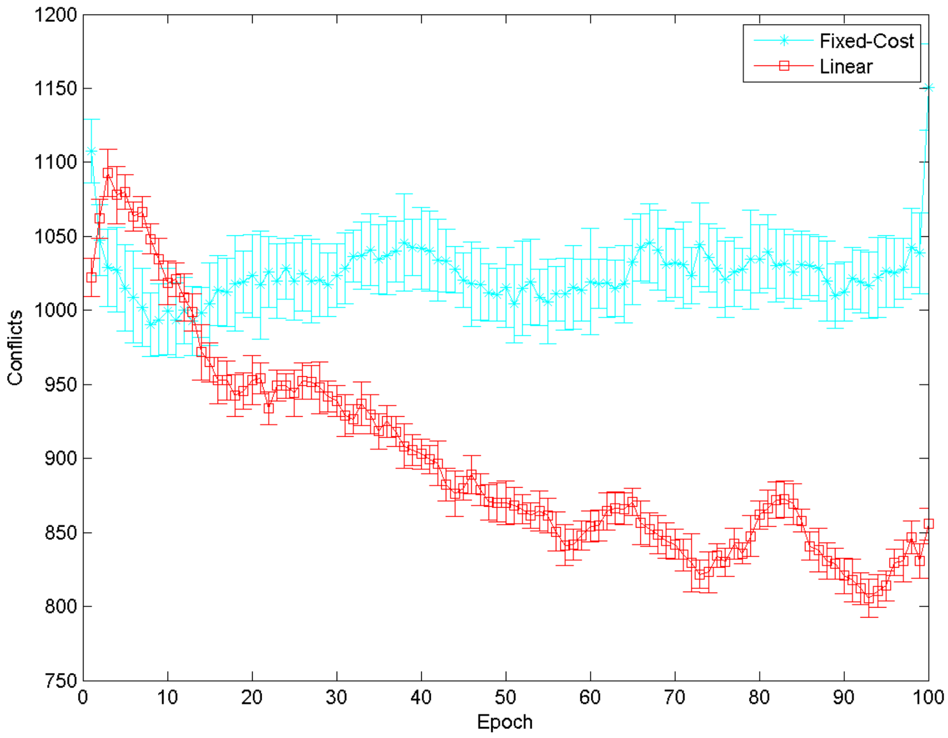


Simulation Experiments

- Urban airspace
  - 256×256 cell map of San Francisco
  - 15 Voronoi partitions
- Fitness calculation
  - Linear: no. conflicts at each cell summed
- UAVs
  - 100 UAVs in airspace during single learning epoch
  - A* planning at both sector- and low-level
  - Conflict radius: 2 cells (approx. 4m)



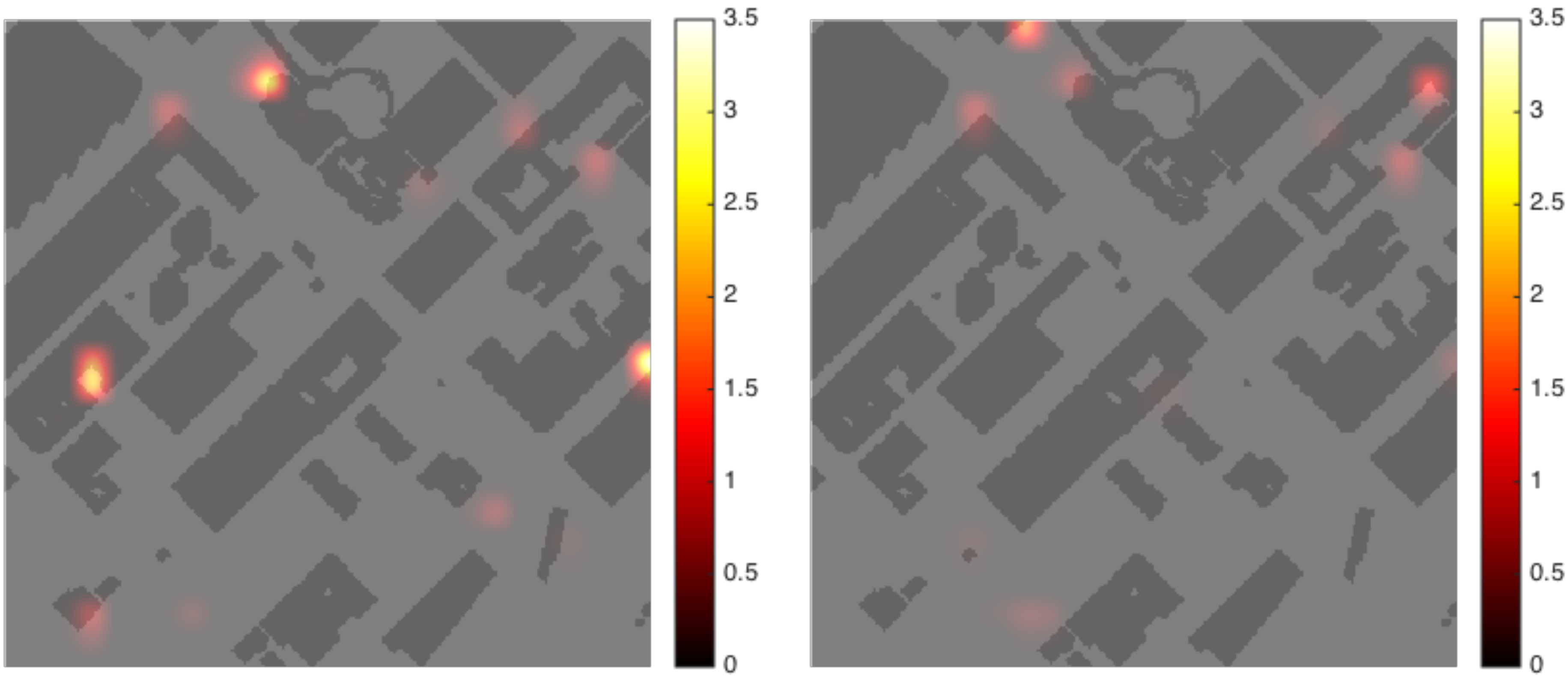
Learning Results: Total Conflicts



- Team performance over 100 learning epochs
- Averaged over 20 trials
- 16% reduction in total system conflicts

Learning Results: Congestion Reduction

Linear Cost Fitness Function



Random initialized sector costs

Learned sector costs

(Rebhuhn et al. IROS 2015, to appear)

Multirobot cooperation

- Future directions
  - Resource-aware coordination
    - System degrades gracefully as computation and communication decreases
    - Scalability to large scale systems
  - Operator-driven objectives
    - Human selects priority of objectives
    - Reduce cognitive load on operator

Questions?

Oregon State UNIVERSITY

Geoffrey A. Hollinger

Robotic Decision Making Laboratory

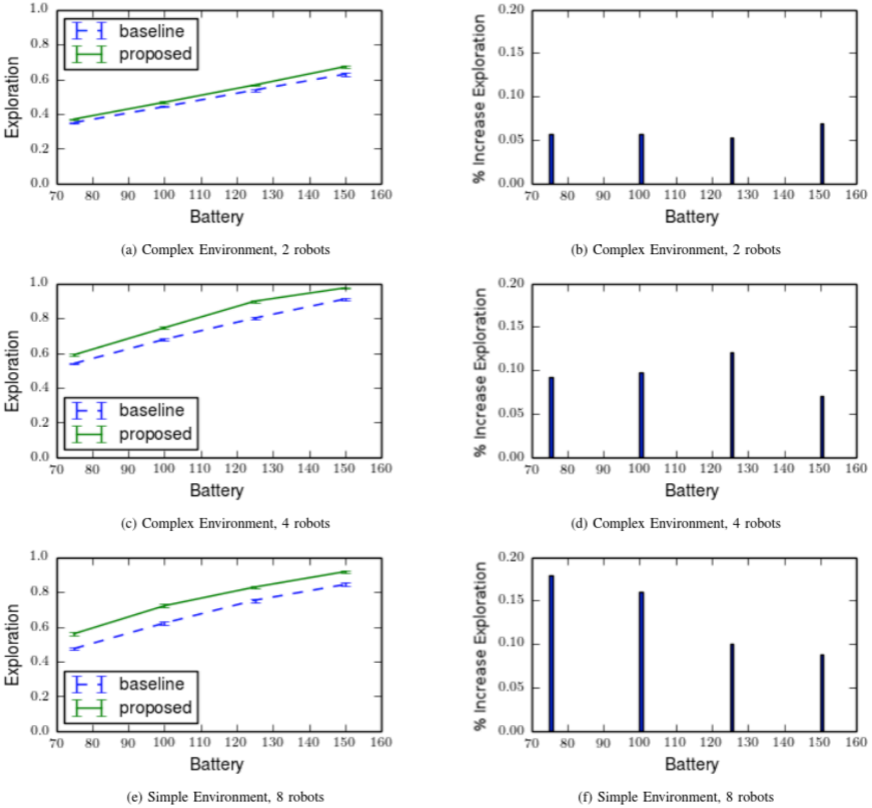
geoff.hollinger@oregonstate.edu

http://research.engr.oregonstate.edu/rdml/

Funding for this work: Office of Naval Research, National Science Foundation, and NASA

Multi-UAV Exploration

- Can use other exploration techniques with state machine on top
- Improvements range from 5% to 18%
- Better results with a larger team



Average of 200 simulation runs, with random start points.