Multi-Robot Learning for Continuous Area Sweeping

Peter Stone

Joint work with **Mazda Ahmadi**
Learning Agents Research Group (LARG)
Department of Computer Sciences
The University of Texas at Austin

LAMAS, July 2005

Multiagent Learning in LARG

Multiagent Learning in LARG

- **Transfer Learning** in Keepaway [Taylor, Wed., 10:30]

Multiagent Learning in LARG

- **Transfer Learning** in Keepaway [Taylor, Wed., 10:30]
- Multiagent **Traffic** Management [Dresner, 10:45]

Multiagent Learning in LARG

- **Transfer Learning** in Keepaway [Taylor, Wed., 10:30]
- Multiagent **Traffic** Management [Dresner, 10:45]
- General Game Playing [Kuhlmann, Dresner]
- Winner, 2005 **RoboCup coach** comp. [Kuhlmann, Knox]

Multiagent Learning in LARG

- **Transfer Learning** in Keepaway [Taylor, Wed., 10:30]
- Multiagent **Traffic** Management [Dresner, 10:45]
- General Game Playing [Kuhlmann, Dresner]
- Winner, 2005 **RoboCup coach** comp. [Kuhlmann, Knox]
- Learning for **Continuous Area Sweeping** [Ahmadi, 2005]

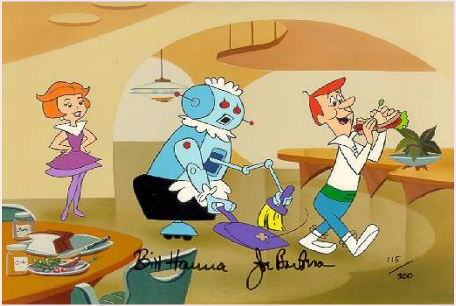
Multiagent Learning in LARG

- **Transfer Learning** in Keepaway [Taylor, Wed., 10:30]
- Multiagent **Traffic** Management [Dresner, 10:45]
- General Game Playing [Kuhlmann, Dresner]
- Winner, 2005 **RoboCup coach** comp. [Kuhlmann, Knox]
- Learning for **Continuous Area Sweeping** [Ahmadi, 2005]
  - Mostly **single-robot**
  - Initial **multi-robot** results

Project Description

Definitions

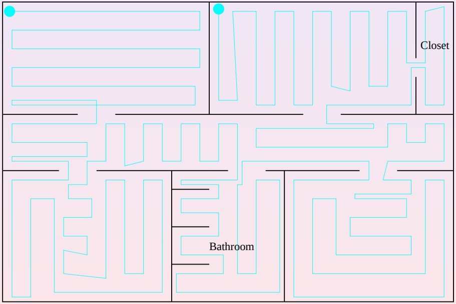
- Area sweeping
- Continuous area sweeping
  - Examples: cleaning robots, surveillance robots.
  - Non-uniform sweeping
  - Multi-robot sweeping



Project Description

Definitions

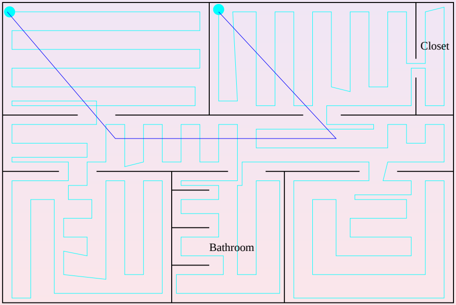
- **Area sweeping**
- Continuous area sweeping
  - Examples: cleaning robots, surveillance robots.
  - Non-uniform sweeping
  - Multi-robot sweeping



Project Description

Definitions

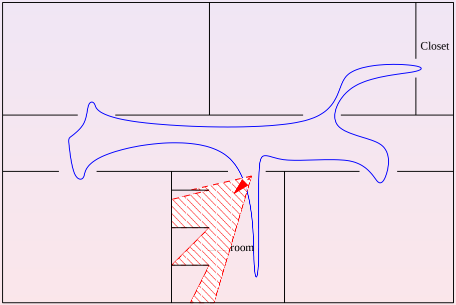
- Area sweeping
- **Continuous area sweeping**
  - **Examples: cleaning robots**, surveillance robots.
  - Non-uniform sweeping
  - Multi-robot sweeping



Project Description

Definitions

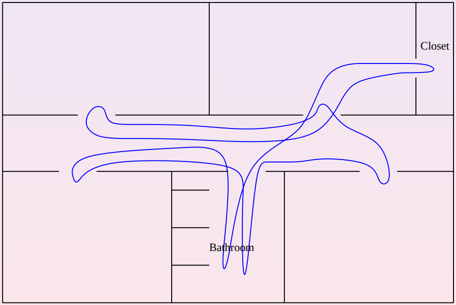
- Area sweeping
- Continuous area sweeping
  - Examples: cleaning robots, **surveillance robots.**
  - Non-uniform sweeping
  - Multi-robot sweeping



Project Description

Definitions

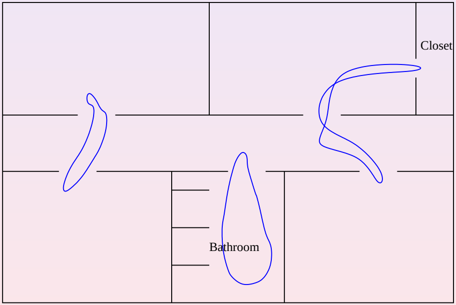
- Area sweeping
- Continuous area sweeping
  - Examples: cleaning robots, surveillance robots.
  - **Non-uniform sweeping**
  - Multi-robot sweeping



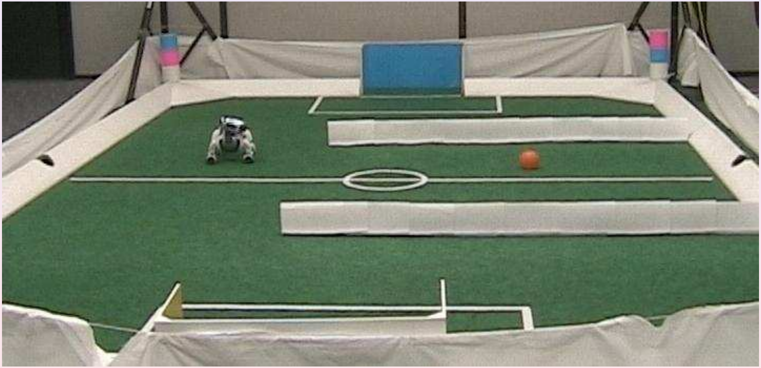
Project Description

Definitions

- Area sweeping
- Continuous area sweeping
  - Examples: cleaning robots, surveillance robots.
  - Non-uniform sweeping
  - **Multi-robot sweeping**



Project Description



Outline

1. Introduction and Motivation

Outline

1. Introduction and Motivation
2. Single Robot Problem Specification

Outline

1. Introduction and Motivation
2. Single Robot Problem Specification
3. Exploration Algorithm
    - Learning Expected Rewards
    - Planning
    - Correctness

Outline

1. Introduction and Motivation
2. Single Robot Problem Specification
3. Exploration Algorithm
   - Learning Expected Rewards
   - Planning
   - Correctness
4. Results
   - Simulation Results
   - Results on Real Robots

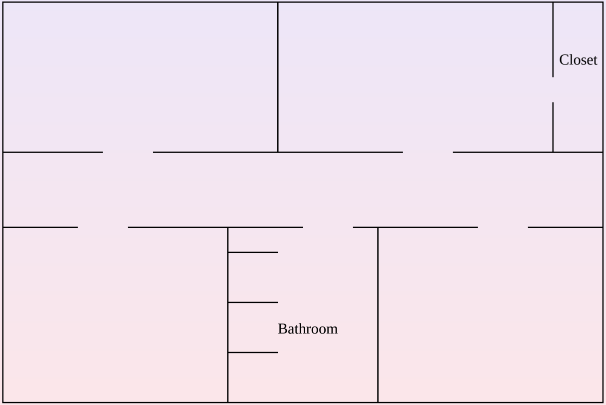
Outline

1. Introduction and Motivation
2. Single Robot Problem Specification
3. Exploration Algorithm
   - Learning Expected Rewards
   - Planning
   - Correctness
4. Results
   - Simulation Results
   - Results on Real Robots
5. Multi-robot Extensions
   - Overview
   - Negotiation Algorithm
   - Results

Outline

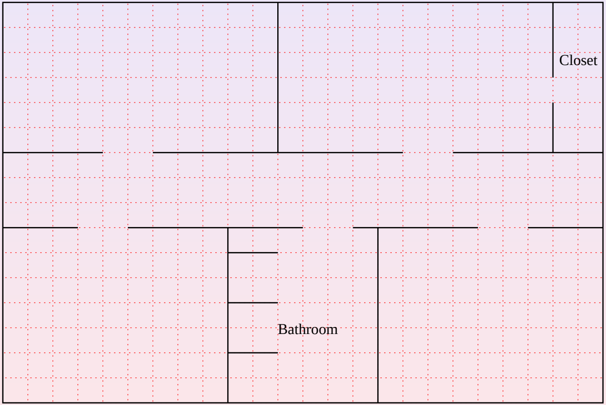
1. Introduction and Motivation
2. Single Robot Problem Specification
3. Exploration Algorithm
   - Learning Expected Rewards
   - Planning
   - Correctness
4. Results
   - Simulation Results
   - Results on Real Robots
5. Multi-robot Extensions
   - Overview
   - Negotiation Algorithm
   - Results

Assumptions



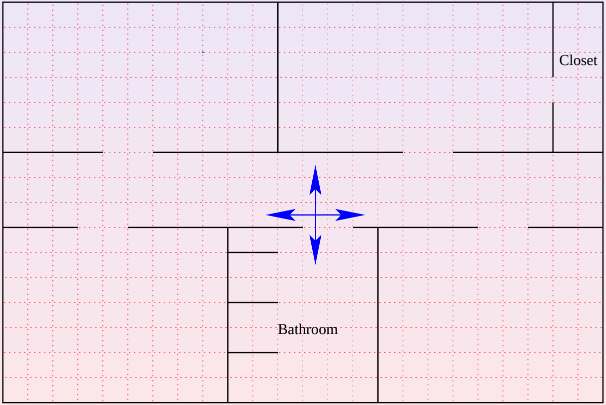
The environment

Assumptions



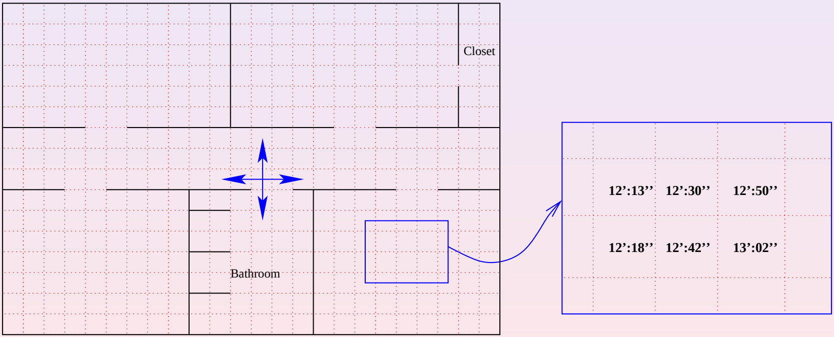
The environment is divided into grid cells (G).

Assumptions



The orientations: *east*, *west*, *north* and *south*.

Assumptions



LV[G]: last time that robot has visited cell g.

Assumptions (cont.)

- Time is considered in sequence of discrete steps.
- imp_e: importance of detecting event e.

Definitions

Formal Definition

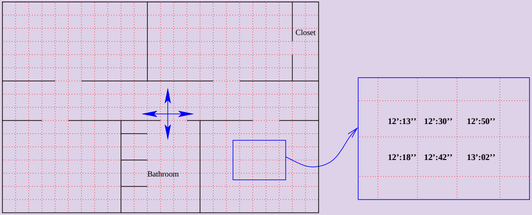
The problem is defined as: $(S, A, T_{sa}, P_{eg}, CF)$:

Definitions

Formal Definition

The problem is defined as: $(\mathbf{S}, A, T_{sa}, P_{eg}, CF)$:

- **S**: Set of states $G \times O \times LV$

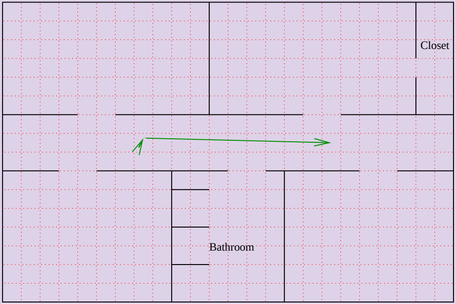


Definitions

Formal Definition

The problem is defined as: $(S, \mathbf{A}, T_{sa}, P_{eg}, CF)$:

- **A**: Set of possible actions

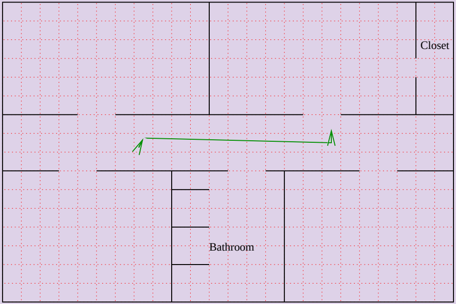


Definitions

Formal Definition

The problem is defined as: $(S, \mathbf{A}\ , T_{sa}, P_{eg}, CF)$:

- **A**: Set of possible actions

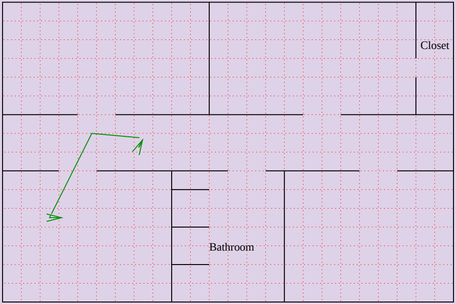


Definitions

Formal Definition

The problem is defined as: $(S, \mathbf{A}, T_{sa}, P_{eg}, CF)$:

- **A**: Set of possible actions

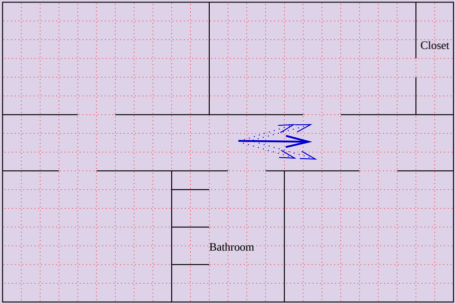


Definitions

Formal Definition

The problem is defined as: $(S, A, \mathbf{T}_{sa}, P_{eg}, CF)$:

- $\mathbf{T}_{sa}$: State transition probabilities

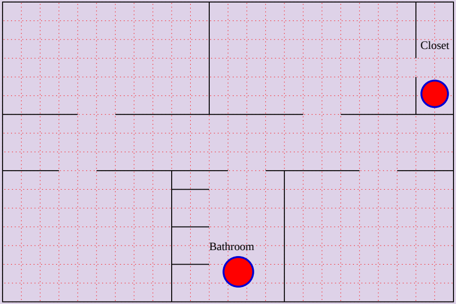


Definitions

Formal Definition

The problem is defined as: $(S, A, T_{sa}, \mathbf{P}_{eg}, CF)$:

- $\mathbf{P}_{eg}$: Probability of appearance of event e in cell g; Initially **unknown**; possibly non-stationary

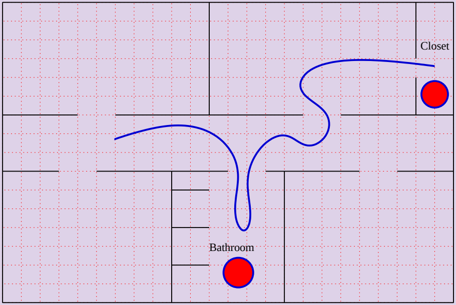


Definitions

Formal Definition

The problem is defined as: $(S, A, T_{sa}, P_{eg}, \mathbf{CF})$:

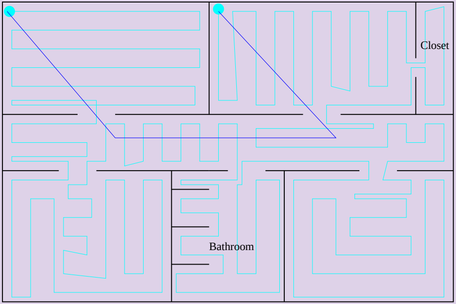
- **CF**: Cost function of the policy. Average time between appearance and detection, weighted by imp_e.



Definitions

The Goal

The goal is to find a policy $\pi : S \to A$ which minimizes the cost function.



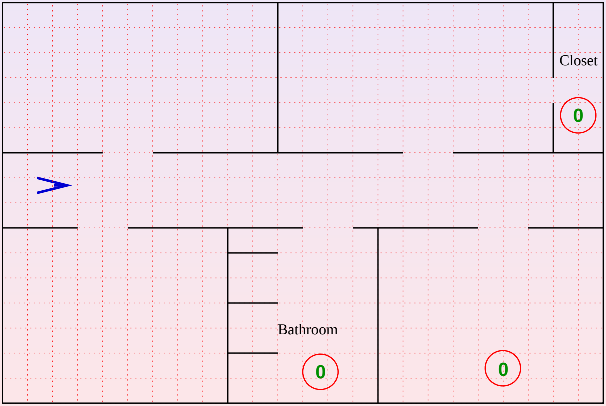
Outline

1 Introduction and Motivation
2 Single Robot Problem Specification
3 Exploration Algorithm
- Learning Expected Rewards
- Planning
- Correctness

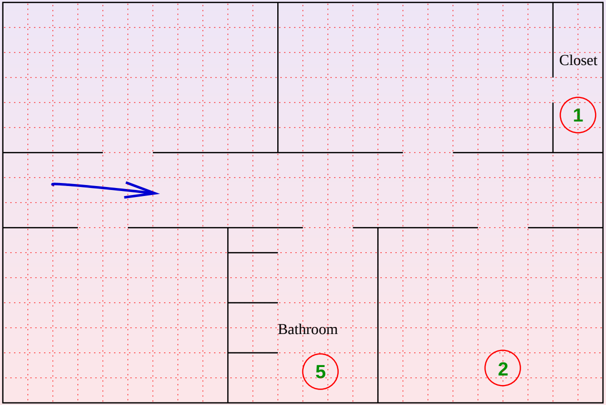
4 Results
- Simulation Results
- Results on Real Robots

5 Multi-robot Extensions
- Overview
- Negotiation Algorithm
- Results

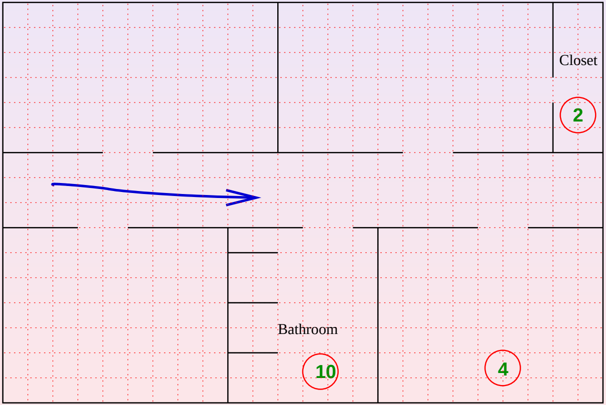
Algorithm Overview



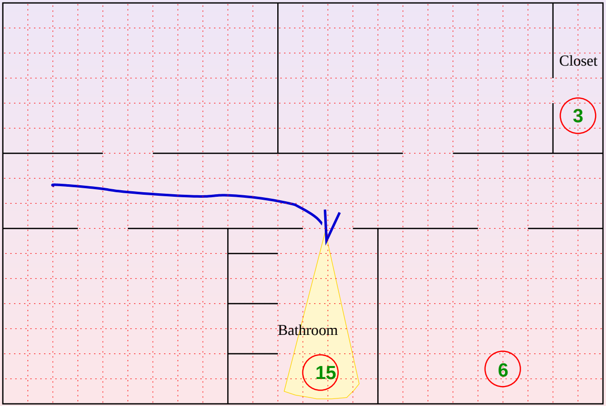
Algorithm Overview



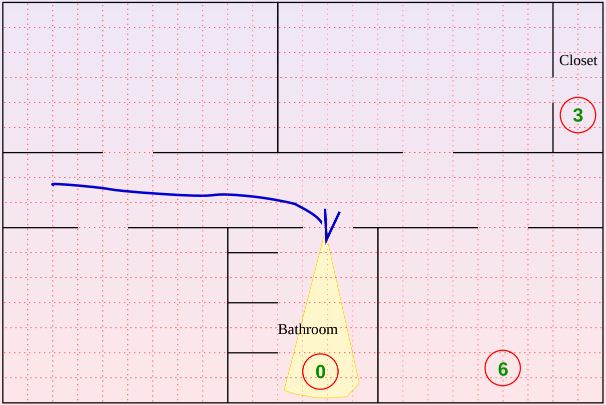
Algorithm Overview



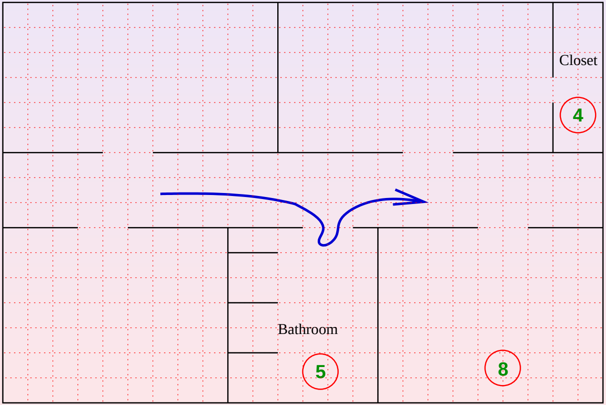
Algorithm Overview



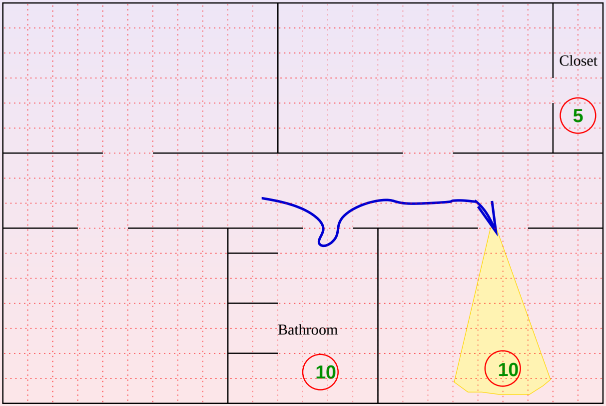
Algorithm Overview



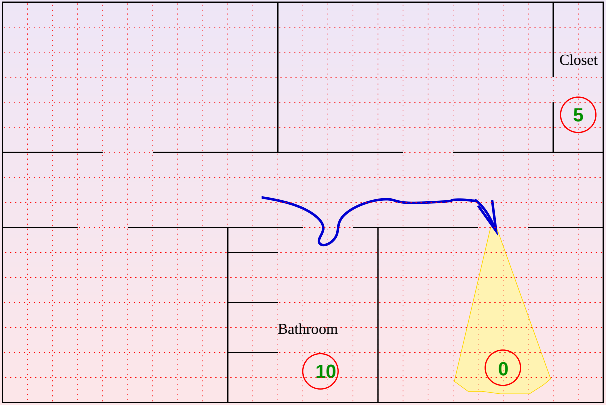
Algorithm Overview



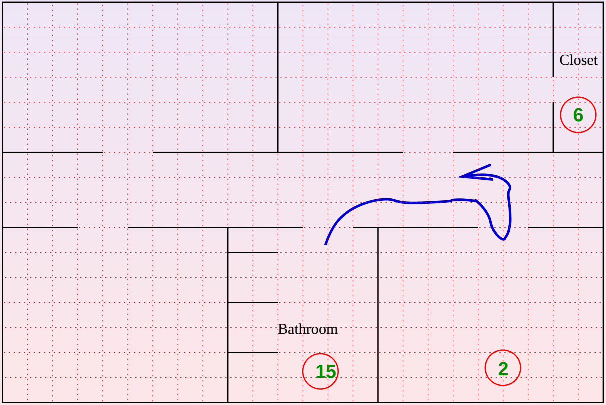
Algorithm Overview



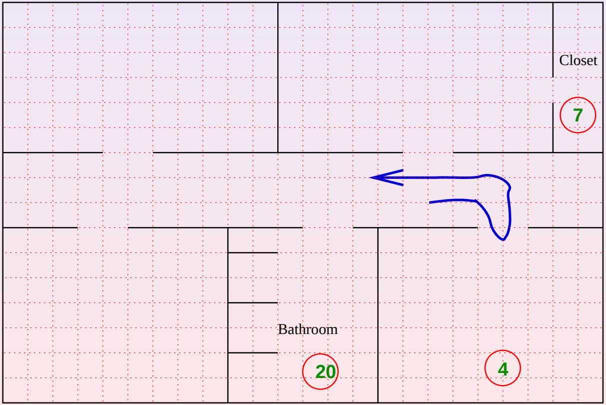
Algorithm Overview



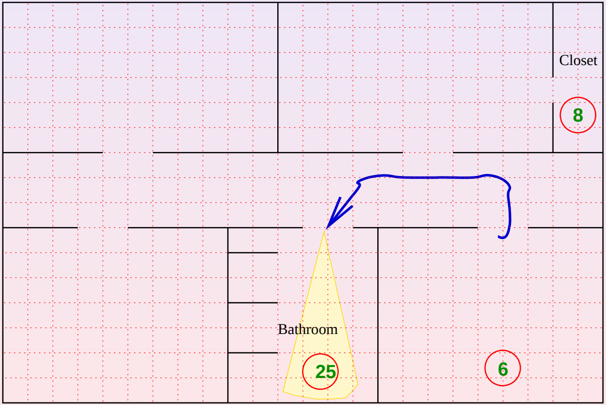
Algorithm Overview



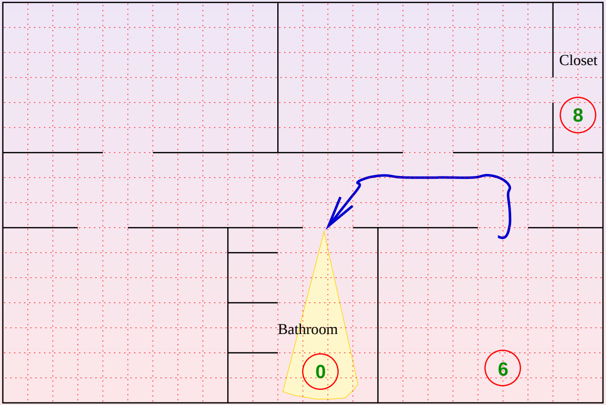
Algorithm Overview



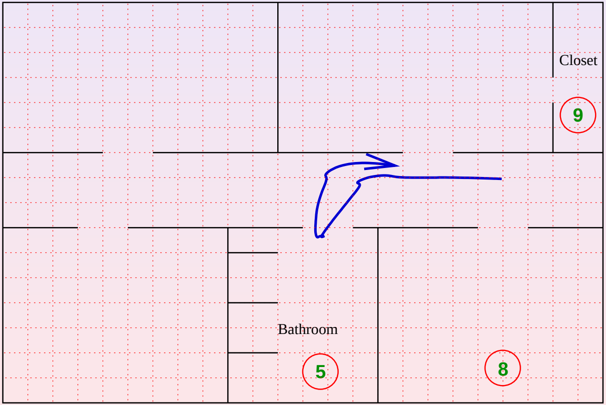
Algorithm Overview



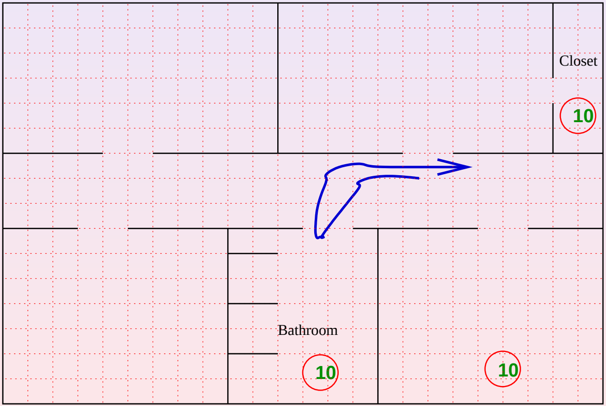
Algorithm Overview



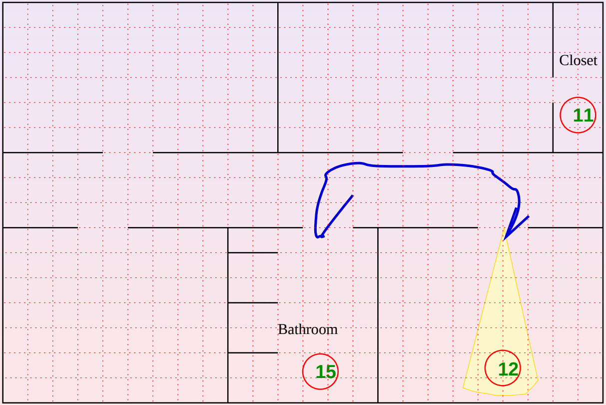
Algorithm Overview



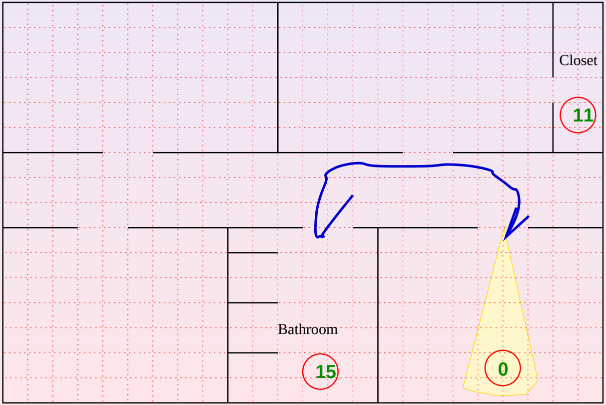
Algorithm Overview



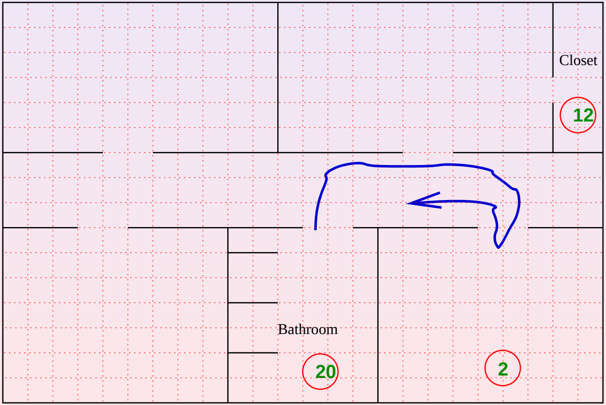
Algorithm Overview



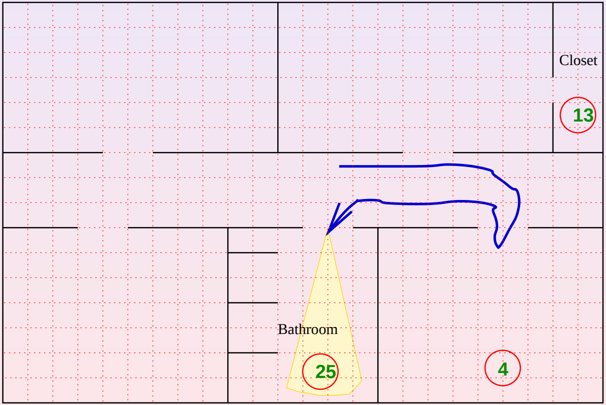
Algorithm Overview



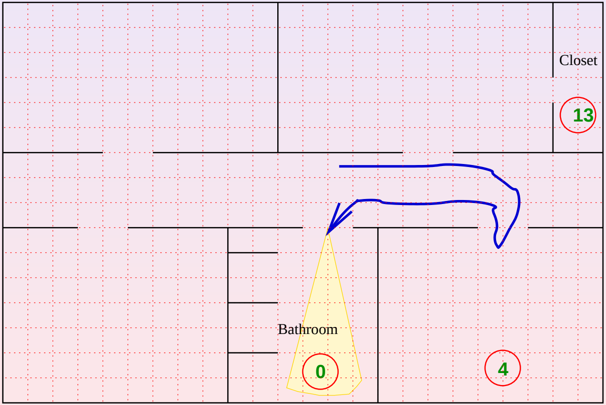
Algorithm Overview



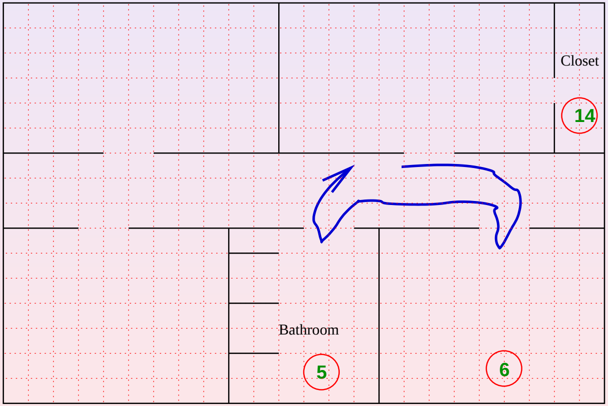
Algorithm Overview



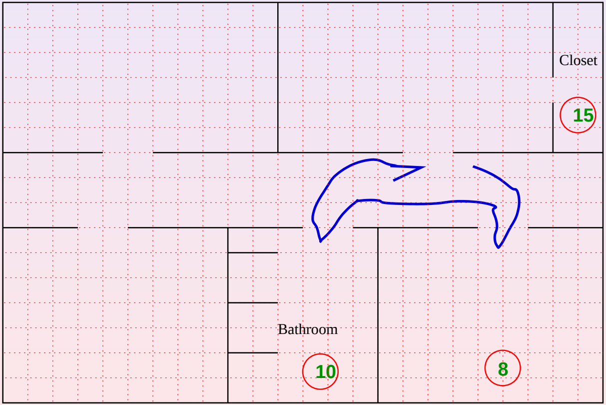
Algorithm Overview



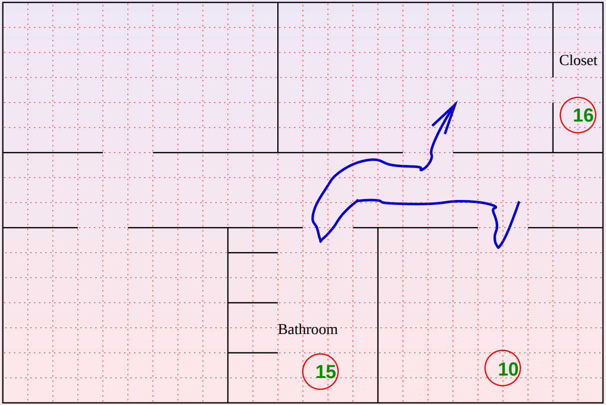
Algorithm Overview



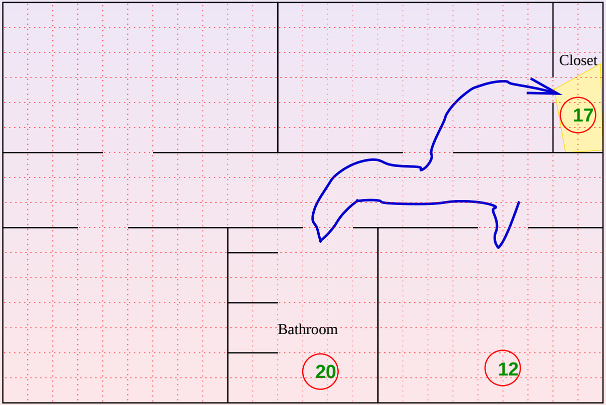
Algorithm Overview



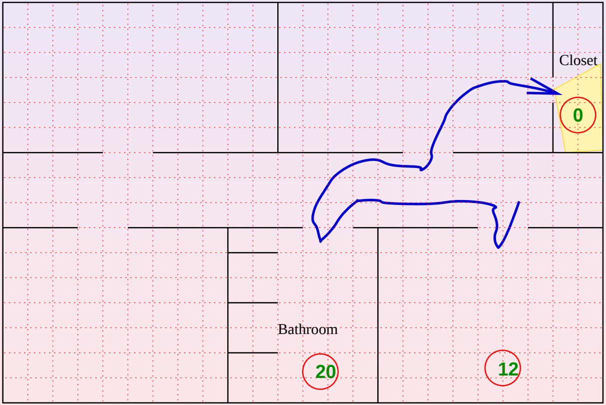
Algorithm Overview



Algorithm Overview



Algorithm Overview



Learning

$$exp\_reward_{gt} = (t - LV[g]) \times \sum_{all\ e} P_{eg} \times imp_e \qquad (1)$$

Learning

$$exp\_reward_{gt} = (t - LV[g]) \times \sum_{all\ e} P_{eg} \times imp_e \qquad (1)$$

$$pot\_reward_{gt} = \sum_{all\ e} P_{eg} \times imp_e \qquad (2)$$

Learning

$$exp\_reward_{gt} = (t - LV[g]) \times \sum_{all\ e} P_{eg} \times imp_e \tag{1}$$

$$pot\_reward_{gt} = \sum_{all\ e} P_{eg} \times imp_e \tag{2}$$

Approximate *pot_reward*

- Compute a new approximation of *pot_reward* (*new_pot*).

Learning

$$exp\_reward_{gt} = (t - LV[g]) \times \sum_{all\ e} P_{eg} \times imp_e \qquad (1)$$

$$pot\_reward_{gt} = \sum_{all\ e} P_{eg} \times imp_e \qquad (2)$$

Approximate *pot_reward*

- Compute a new approximation of *pot_reward* (*new_pot*).
- $pot\_reward := \alpha \times new\_pot + (1 - \alpha) \times pot\_reward$

Learning

$$exp\_reward_{gt} = (t - LV[g]) \times \sum_{all\ e} P_{eg} \times imp_e \qquad (1)$$

$$pot\_reward_{gt} = \sum_{all\ e} P_{eg} \times imp_e \qquad (2)$$

Approximate *pot_reward*

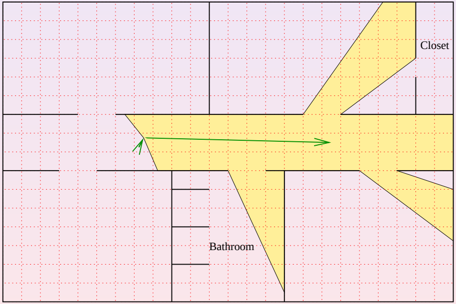
- Compute a new approximation of *pot_reward* (*new_pot*).
- $pot\_reward := \alpha \times new\_pot + (1 - \alpha) \times pot\_reward$
- No updates to zero, instead decay over time.

Planning

- One step greedy action selection
- Set of actions: going to different grids with one of the four orientations.
- What to maximize: Sum of collected *expected rewards* per time.

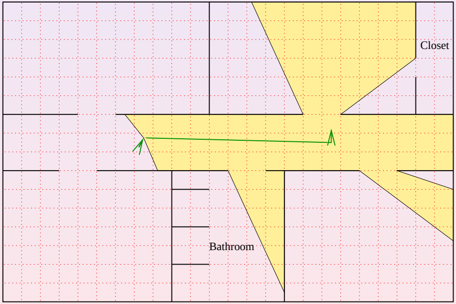
Planning

- One step greedy action selection
- Set of actions: going to different grids with one of the four orientations.
- What to maximize: Sum of collected *expected rewards* per time.



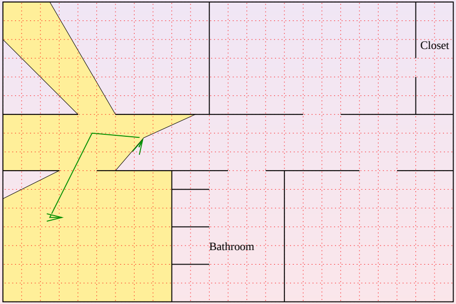
Planning

- One step greedy action selection
- Set of actions: going to different grids with one of the four orientations.
- What to maximize: Sum of collected *expected rewards* per time.



Planning

- One step greedy action selection
- Set of actions: going to different grids with one of the four orientations.
- What to maximize: Sum of collected *expected rewards* per time.



Correctness Proof

- With *optimal planning*, the cost function is minimized
  - Maximizing exp_reward at individual cells minimizes CF

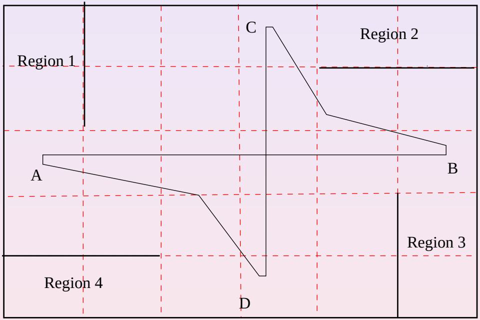
Correctness Proof

- With *optimal planning*, the cost function is minimized
  - Maximizing exp_reward at individual cells minimizes CF
- Formal proof in [Ahmadi & S, 2005]

Outline

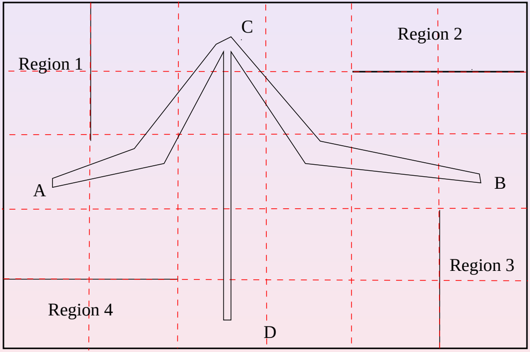
1. Introduction and Motivation
2. Single Robot Problem Specification
3. Exploration Algorithm
   - Learning Expected Rewards
   - Planning
   - Correctness
4. Results
   - Simulation Results
   - Results on Real Robots
5. Multi-robot Extensions
   - Overview
   - Negotiation Algorithm
   - Results

Simulation Results



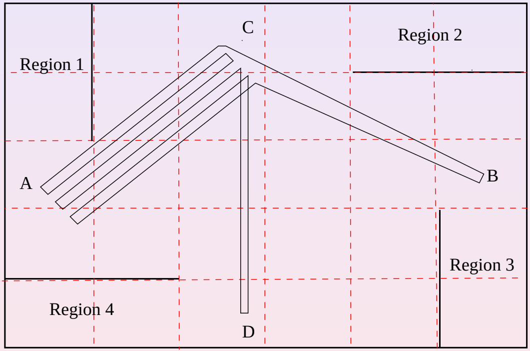
The path that the robot traverses in uniform distribution of the appearance of the ball. **Average detection time: 106 seconds.**

Simulation Results (cont.)



The path the robot traverse when the ball always appears in region 2. **Average detection time: 47 seconds.**

Simulation Results (cont.)



Biased distribution: Probability of the ball appearance is 60% in region 2, 30% in region 1 and 5% in region 3 and 4. **Average detection time: 79 seconds.**

Simulation Results (cont.)

Changing Distribution

From the previous distribution to uniform distribution, it took about 9 loops to adapt the correct distribution.

Results from Real Robots

Movies!

Outline

1. Introduction and Motivation
2. Single Robot Problem Specification
3. Exploration Algorithm
   - Learning Expected Rewards
   - Planning
   - Correctness
4. Results
   - Simulation Results
   - Results on Real Robots
5. Multi-robot Extensions
   - Overview
   - Negotiation Algorithm
   - Results

Problem Overview

- Multiple robots **divide** the sweeping area
- Goal: minimize global cost function (**fully cooperative**)

Problem Overview

- Multiple robots **divide** the sweeping area
- Goal: minimize global cost function (**fully cooperative**)
  - Equalized (weighted) average detection time among robots

Problem Overview

- Multiple robots **divide** the sweeping area
- Goal: minimize global cost function (**fully cooperative**)
  - Equalized (weighted) average detection time among robots

- **Team members** change dynamically
  - Robots regularly added and removed

Problem Overview

- Multiple robots **divide** the sweeping area
- Goal: minimize global cost function (**fully cooperative**)
  - Equalized (weighted) average detection time among robots

- **Team members** change dynamically
  - Robots regularly added and removed
- P_{eg}'s still change dynamically

Solution Framework

- Robots each use **single-agent algorithm** in limited region
- Continual **negotiation** at region boundaries

Solution Framework

- Robots each use **single-agent algorithm** in limited region
- Continual **negotiation** at region boundaries
- **New robots** take minimal area in immediate neighborhood
- Area of **removed robot** initially taken by neighbor

Negotiation Algorithm Sketch

1. Periodically **communicate** visit intervals for boundary cells

Negotiation Algorithm Sketch

1. Periodically **communicate** visit intervals for boundary cells
2. Consider “**taking over**” neighbor’s worst cell
   - Compute **hypothetical plans**, report visit intervals

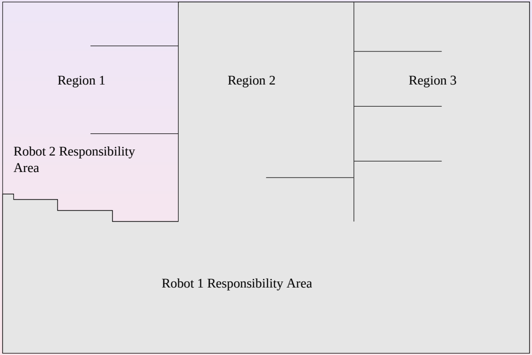
Negotiation Algorithm Sketch

1. Periodically **communicate** visit intervals for boundary cells
2. Consider “**taking over**” neighbor’s worst cell
    - Compute **hypothetical plans**, report visit intervals
3. **Single** best neighboring offer accepted
    - biggest **coverage improvement**

Negotiation Algorithm Sketch

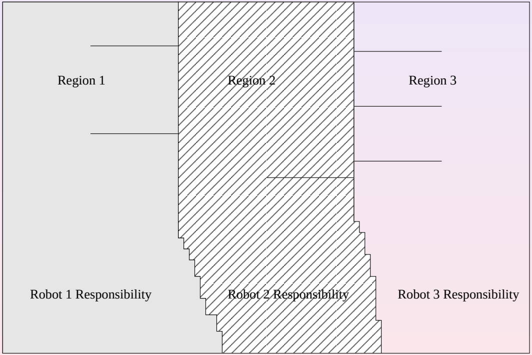
1. Periodically **communicate** visit intervals for boundary cells
2. Consider “**taking over**” neighbor’s worst cell
   - Compute **hypothetical plans**, report visit intervals
3. **Single** best neighboring offer accepted
   - biggest **coverage improvement**
4. Repeat next cycle

Simulation Configuration I



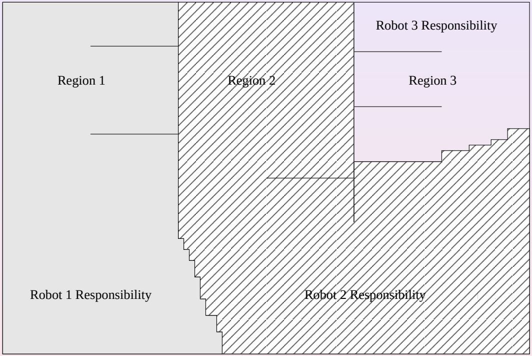
2 homogeneous robots, uniform P_{eg}'s

3 homogeneous robots



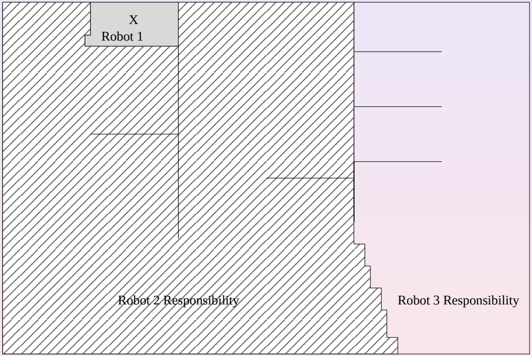
Uniform P_{eg}'s

3 heterogeneous robots



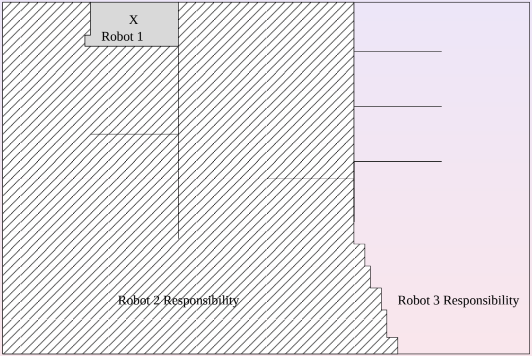
- Robot 3 moves at half speed
- Time between visits, before negotiation: 54s, after:50s.

3 homogeneous robots, non-uniform P_{eg}'s



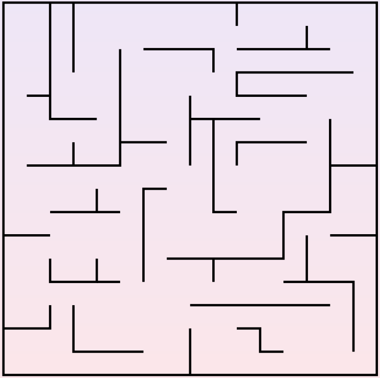
- P_{ex} 10 times greater
- Average detection time, before negotiation: 48s, after: 32s.

3 homogeneous robots, non-uniform P_{eg}'s

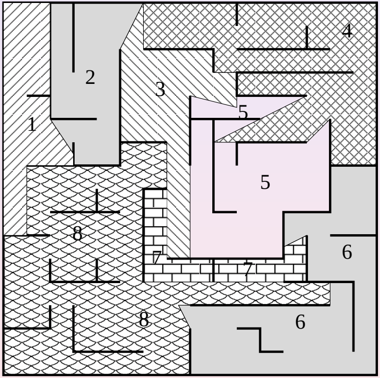


- P_{ex} 1000 times greater
- Average detection time, before negotiation: 48s, after: 1s.

Simulation Configuration II



8 heterogeneous robots



Robot speeds differ from 10 (1 & 3) to 50 (8)

Results from Real Robots

Movie!

Related Work

- Kalra, Stentz, and Ferguson, **Hoplites: A market framework for complex tight coordination in multi-agent teams**, Robotics Institute, CMU
- Kurabayashi and Ota, **Cooperative sweeping by multiple mobile robots**, ICRA 1996
- Choset, **Coverage for robotics; a survey of recent results**, Annals of Math. and AI, 2001.
- Parker, **Distributed algorithms for multi-robot observation of multiple moving targets**, Autonomous Robots, 2002.
- Koenig, Szymanski, and Liu. **Efficient and Inefficient Ant Coverage Methods.** Annals of Math. and AI, 2001

Conclusion and Future Work

Conclusion

Continuous area sweeping interesting and challenging.
Good initial progress

Conclusion and Future Work

Conclusion

Continuous area sweeping interesting and challenging.
Good initial progress

Future Work

- Non-greedy planning
- Continuous representations
- Better representation and analysis of noise
- Reasoning about communicative connectivity

Acknowledgements

- Joint work with **Mazda Ahmadi**

Acknowledgements

- Joint work with **Mazda Ahmadi**
- Built on UT Austin Villa robot soccer code
  - **Kurt Dresner, Peggy Fidelman, Nate Kohl**
  - **Greg Kuhlmann, Mohan Sridharan, Dan Stronger**
  - And others