Distributed On-Line Coordination for Multi-Robot Patrolling

L. Iocchi[1], A. Farinelli[2], D. Nardi[1]

[1] DIAG, Sapienza University of Rome, Italy

[2] University of Verona, Italy

Motivation

RoCoCo **Cognitive Cooperating Robots** Lab.

Research on Multi-robot systems

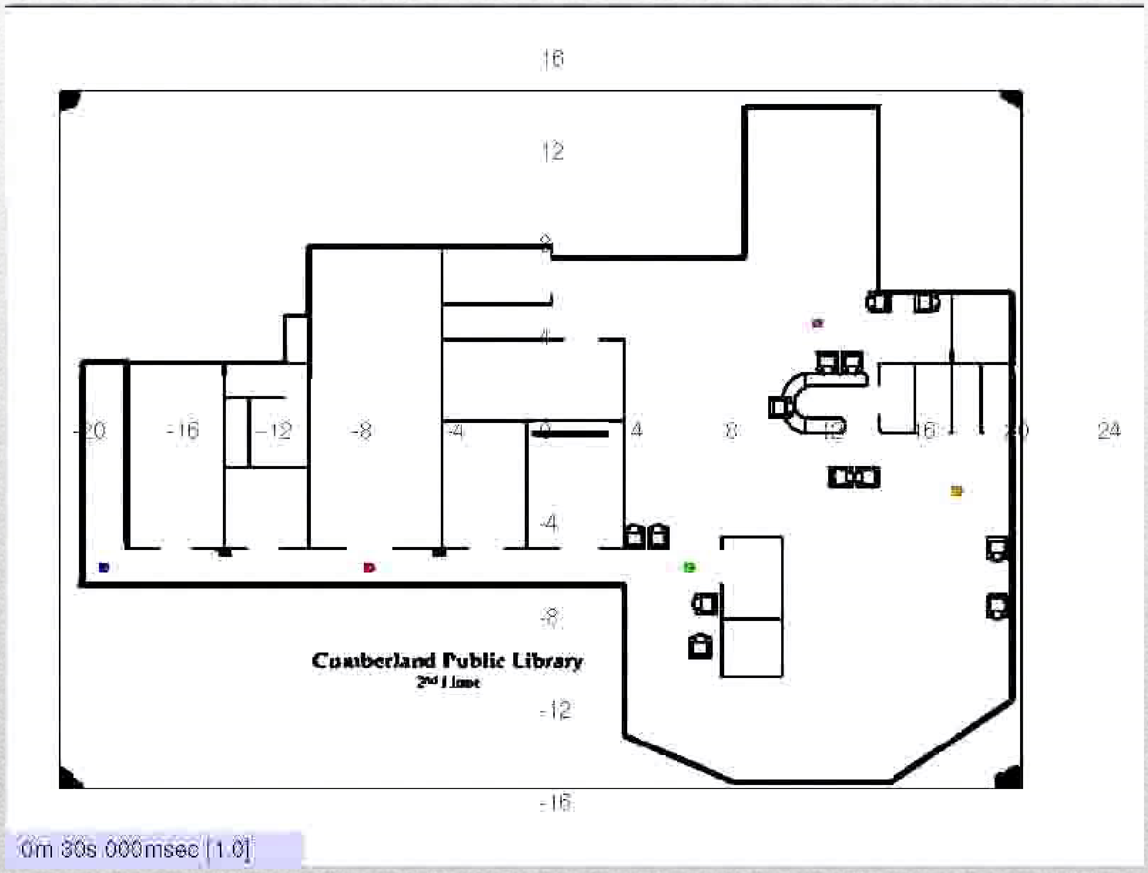
Motivation

Research projects on Multi-robot surveillance (mobile sensor network)

- **Bee-SAFE - Multi-robot mapping and exploration**, (2011-2013), Sistemi Software Integrati, SELEX-SI
- **Smart Monitoring of Complex Public Scenes**, (2011-2013), Dept. of Homeland Security (DHS), USA
- **Situation Awareness**, Iniziativa Software 2, (2010-2012), SESM, SELEX-SI
- **Multi-Robot Teams for Environmental Monitoring**, (2009-2011), Dept. of Homeland Security (DHS), USA
- **SAMAS - Adaptive multi-robot services and its applications to demining**, (2009- 2011), Space Software Italia.

Multi-Robot Patrolling

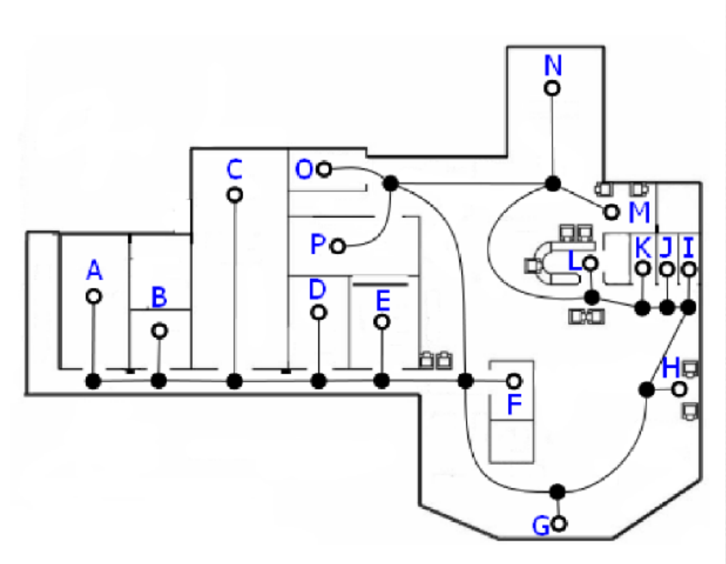
Continuous monitoring of a (known) environment



MRP state of the art

- Graph based representation
- Performance metrics based on idleness and frequency
- Perfect perception, action and communication

- Optimal **off-line centralized** solutions
  - (e.g., derived using graph theory [Portugal and Rocha 11])



MRP our approach

- Graph based representation
- Performance metrics based on idleness and frequency
- Possible failures, robot interferences
- (Suboptimal) **on-line distributed** solutions
  - Based on Dynamic Task Assignment
- Non-adversarial environments
  - Visit places as often as possible

MRP Formulation

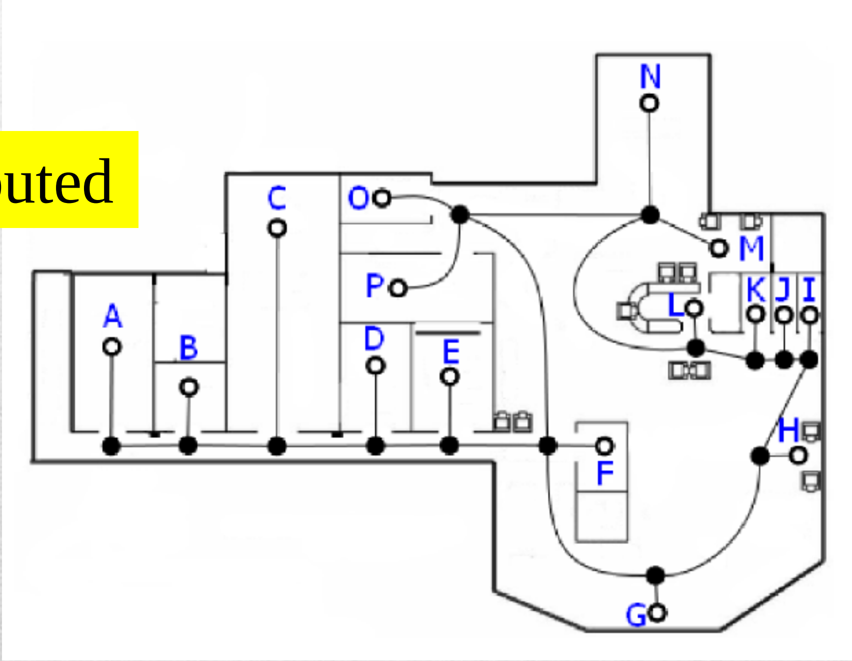
Patrol Graph $PG = < P, E, c >$

P: Poses, E: Edges c: Travel cost

Objective On-line and distributed

Find paths for each robot to minimize performance metrics

Idleness: $I^p(t)$ time between two consecutive visits of the same node



Dynamic Task Assignment for MRP

Assign robot **i** to task (location) **j** → $a_{ij} = 1$

Objective: $$\arg\max_{\mathbf{A}} \sum_{i=1}^{|R|} \sum_{j=1}^{|P|} v_{ij} a_{ij}$$

Reward of allocating **i** to **j**: $v_{ij} = U(r_i, p_j, t)$

$$U(r_i, p_j, t) = \theta_1 I^{p_j}(t) + \theta_2 Tc(r_i, p_j, t)$$

Travel Cost

Subject to:

- one robot per task → avoid conflicts
- one task per robot → avoid planning ahead

DTA Solutions

We aim at a completely **decentralized** solution:

- Each robot maintains $I^p(t)$ for all nodes
- Robots share info on idleness and merge such info
  - Merge: always keep lowest value for each room
- General solution scheme based on events
  - Robot reaches a visit location
  - Robot receives a message
- Two solutions
  - DTA-Greedy (always go to best location and deconflict)
  - DTAP (Sequential Single Item auctions and Partitioning)

DTA-Greedy

<u>When a robot reaches a location:</u>

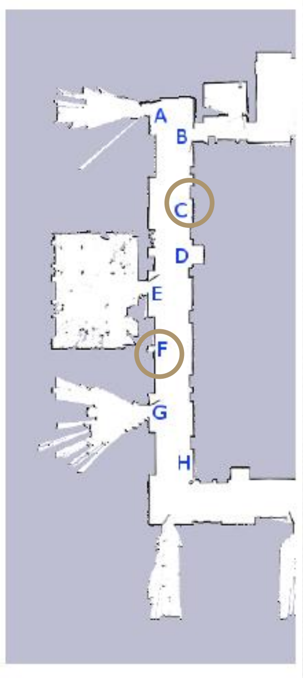
choose next <u>best</u> location (p*)

set idleness of p* to 0 and broadcast

<u>When receiving a message</u> (idleness)

merge info on idleness

revise destination



$$U(r_k, p', t) = \theta_1 I^{p_j}(t) + \theta_2 Tc(r_i, p_j, t) + \theta_3 ps(r_k, p')$$

<u>Priority for location p’</u>

- Defined off-line, easy to set on simple maps
- Balance work-load: e.g., low value for shared nodes

DTA with SSI and Partitioning

When a robot reaches a location:

identify next best location (p*)

announce bid for target p* (b)

collect all bids

if b is not the lowest, select next best location

else consider robot responsible for p*, goto p*

When receiving a message [idleness, bid, announce]

[idleness] merge info on idleness

[bid] updates bids

[announce] compute and send bid for target

DTAP bids

Bids based on utility function $U(r_i, p_j, t) = \theta_1 I^{p_j}(t) + \theta_2 Tc(r_i, p_j, t)$

Cost based on a variable set of tasks assigned to a robot

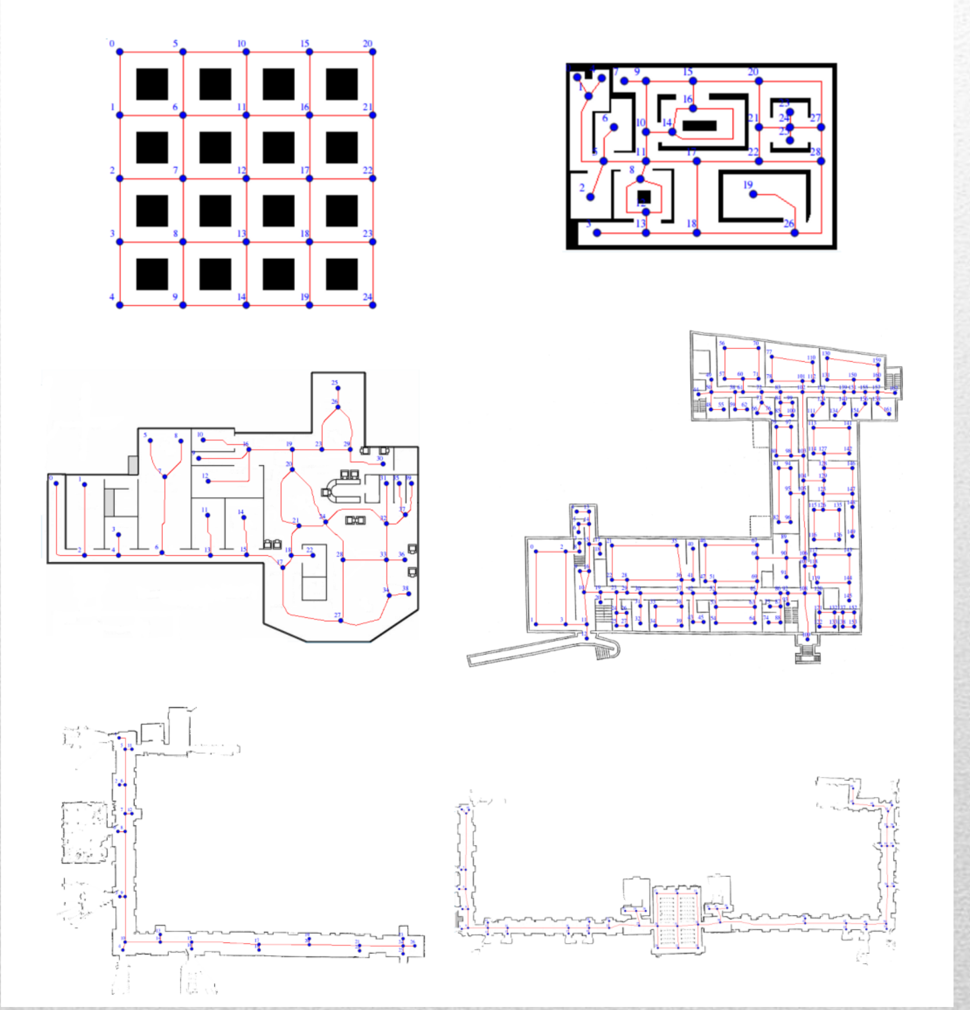
Variable set of tasks determine a **partition** of the graph

Difference with standard auctions:

- no auctioneer
- tasks are not created but selected
- robot takes a task if no one else is better
- time-out when collecting bids (failures)

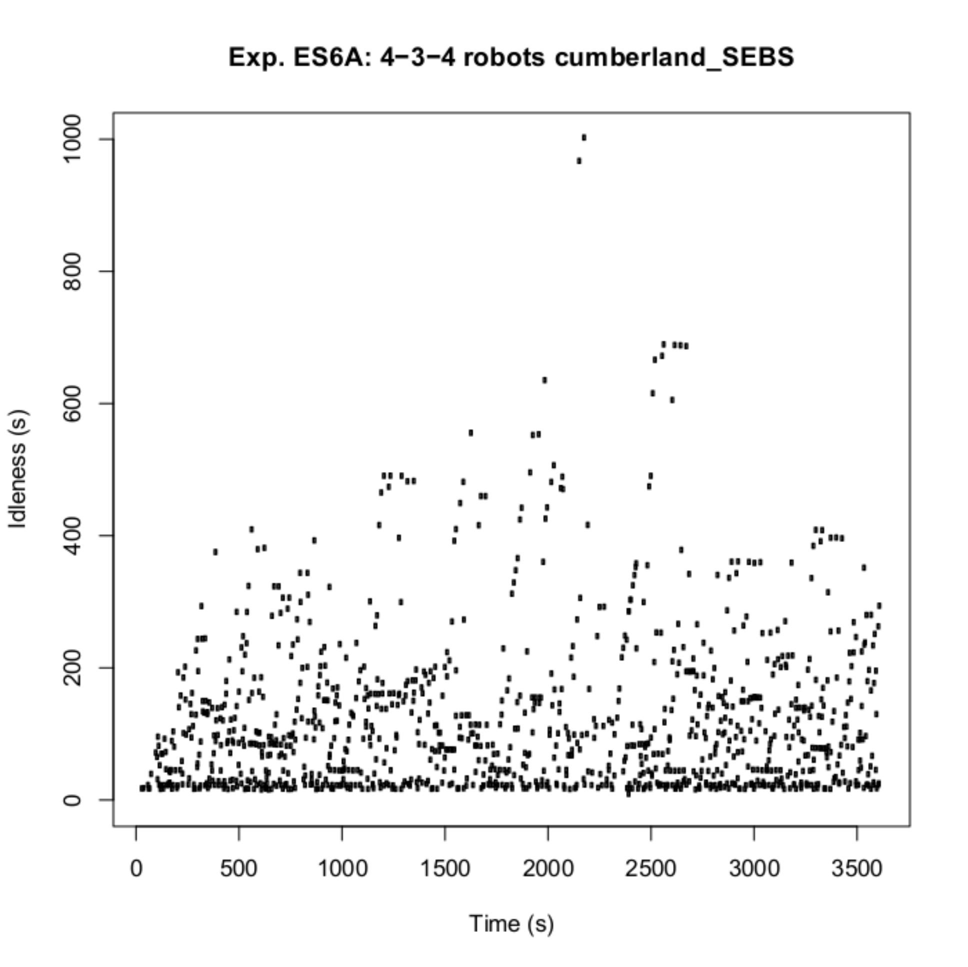
Experimental Set-up

- **Realistic simulator** (ROS & Stage)
- Six maps including real maps of our Dept.
- Many algorithms compared
- New performance representation and analysis



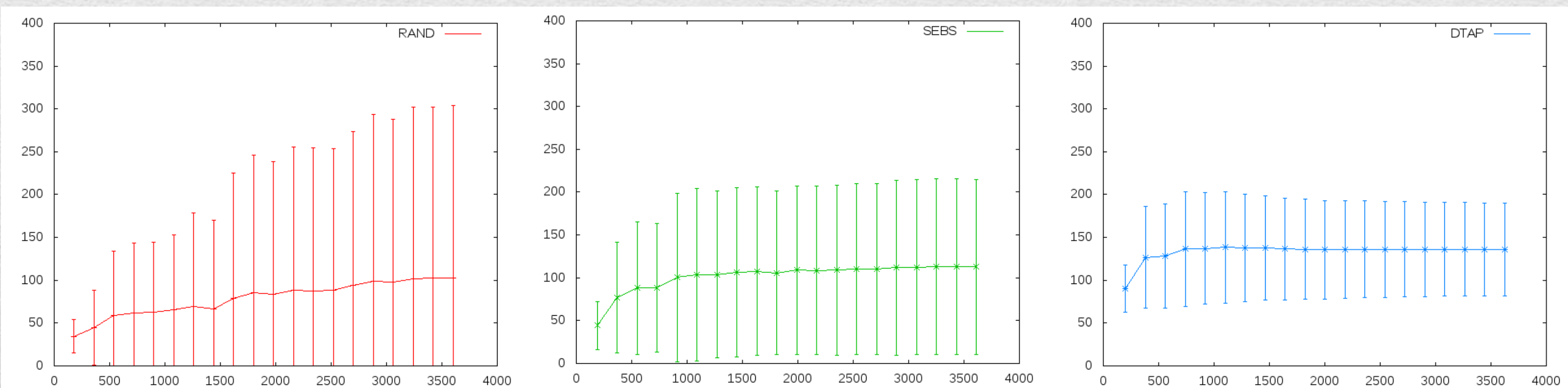
Performance metrics

Idleness: time between two consecutive visits of the same node by any robot in the team



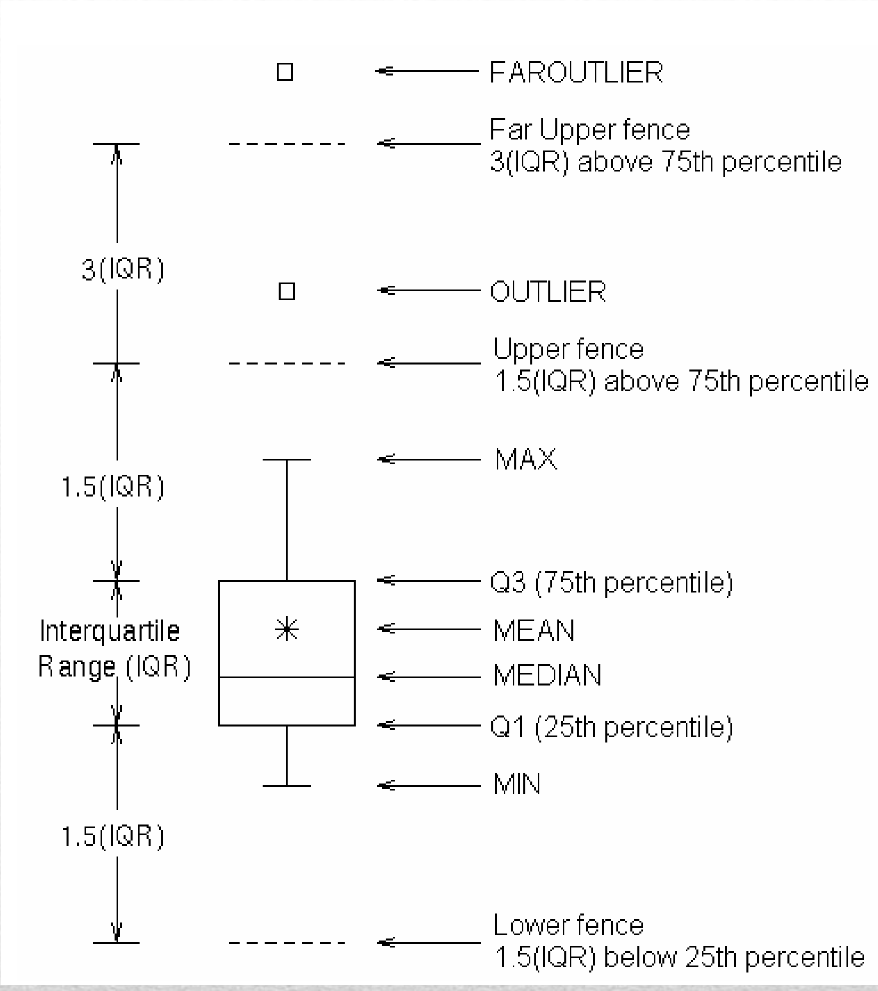
Performance metrics

Idleness avg alone is not a good performance metric
Idleness avg and **stddev** are independent metrics



Performance metrics

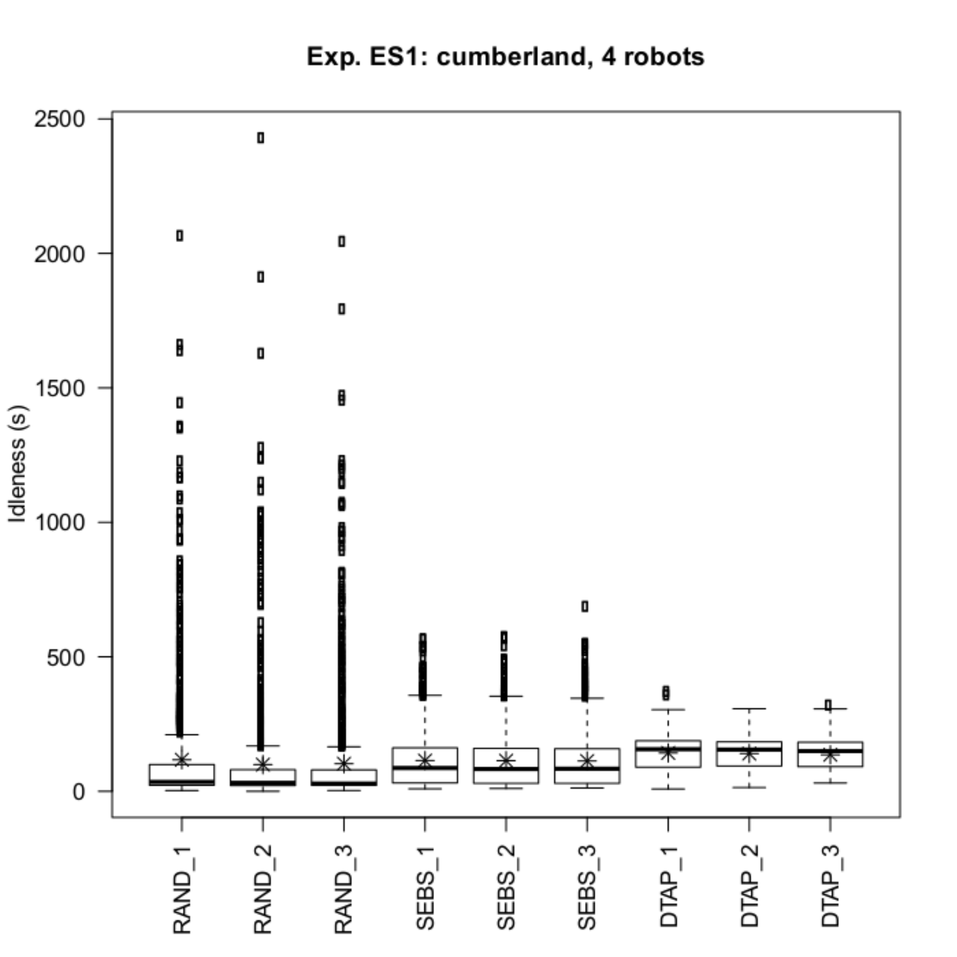
Box-plot representation: Graphical representation for distribution of idleness values during an experiment



Results with simulator

Boxplots

- average
- stddev
- maximum
- outliers

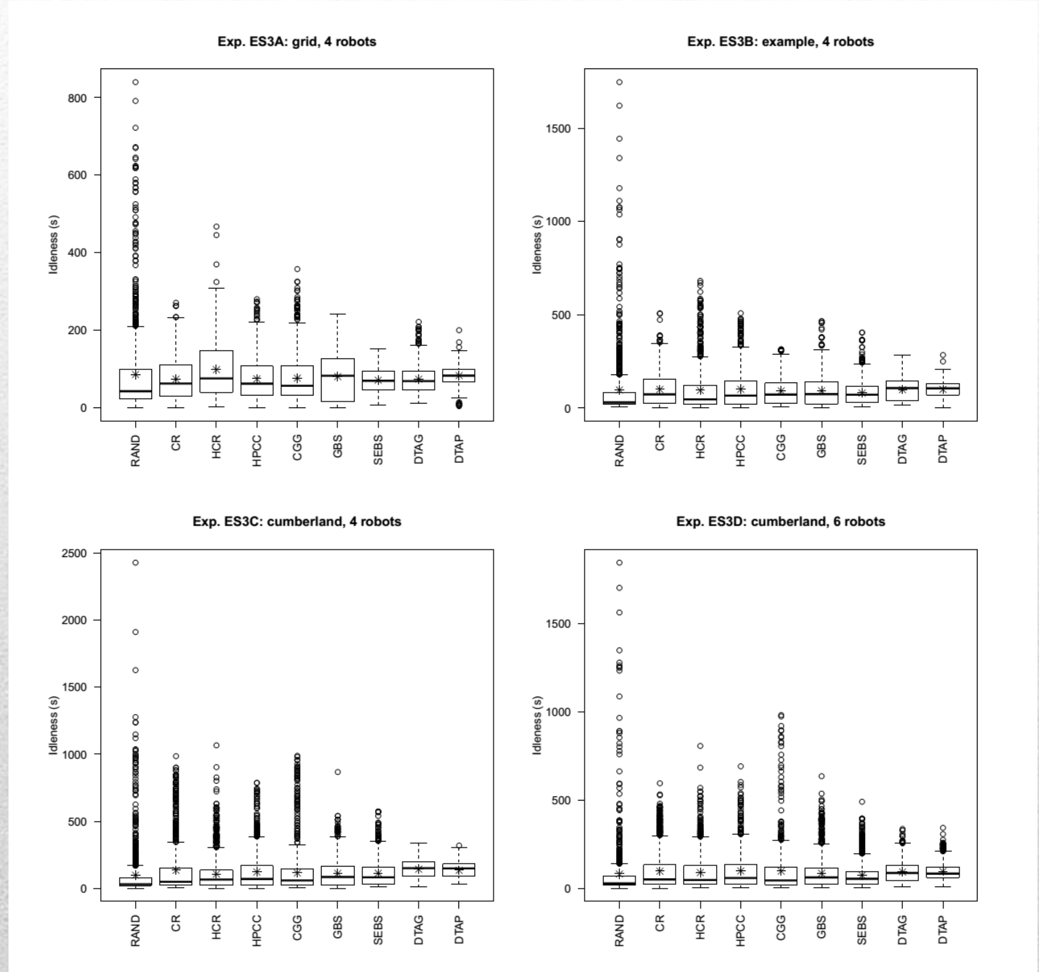


Results with simulator

Comparison

- 9 algorithms
  (off-line vs. on-line)

- Best performance for on-line alg.
  minimize
  avg/stddev idl.
  outliers

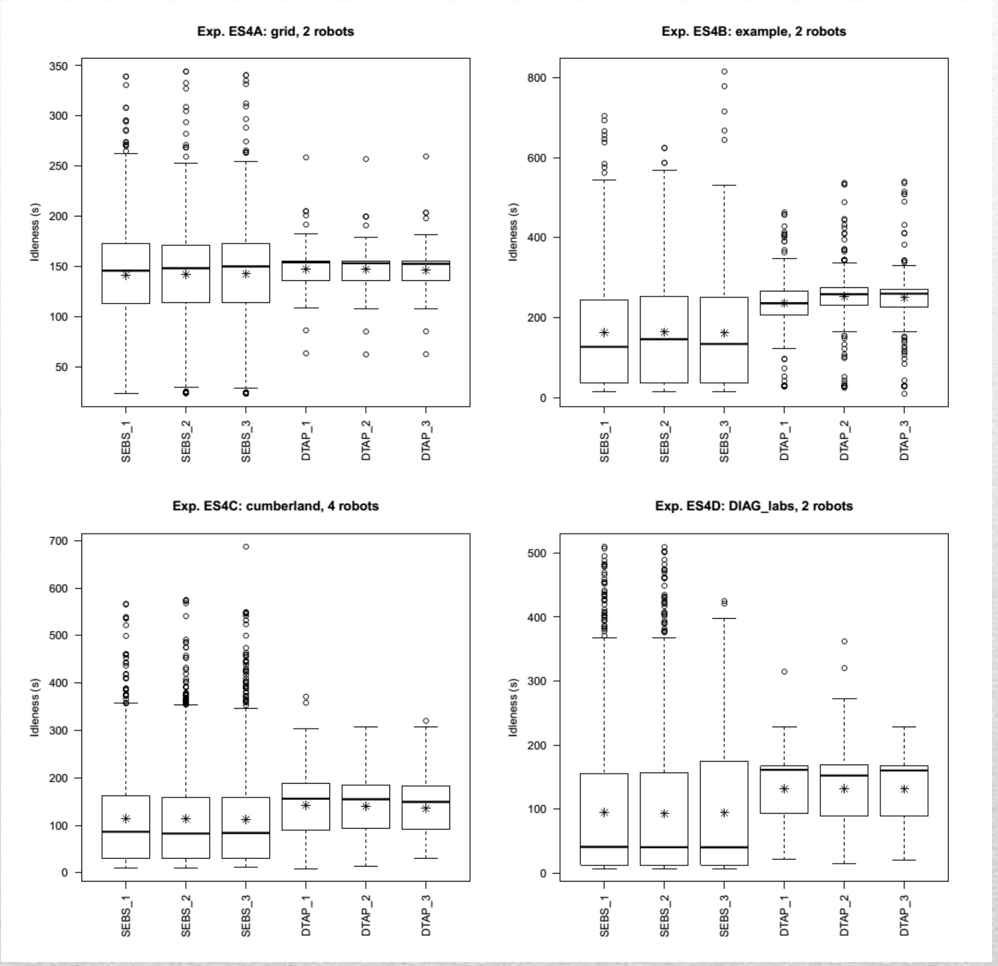


Results with simulator

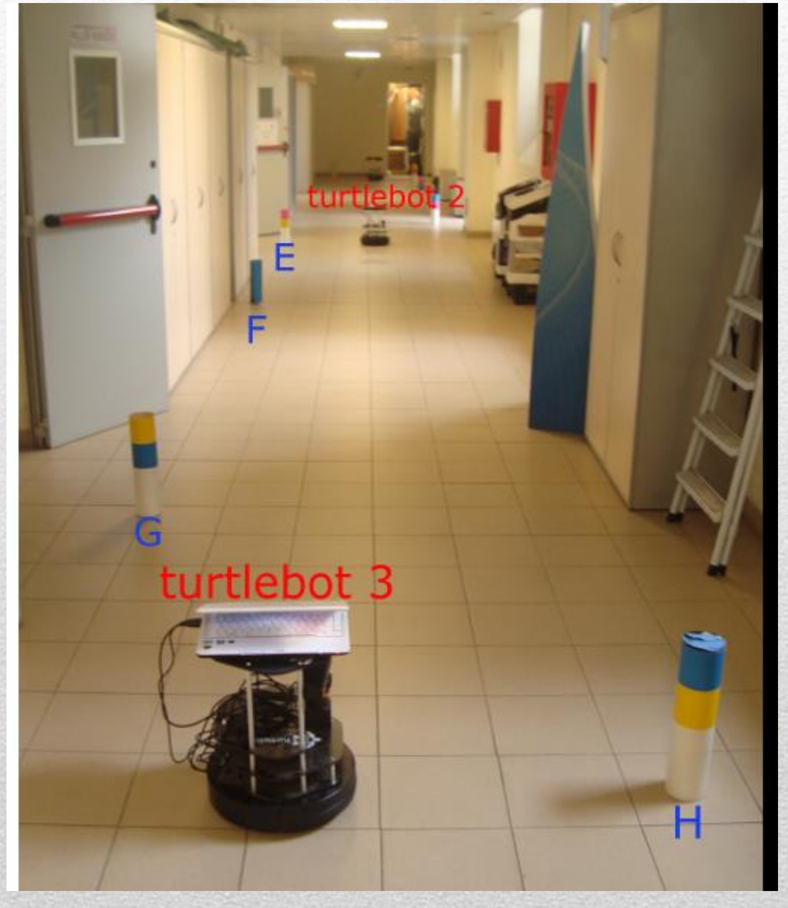
Comparison

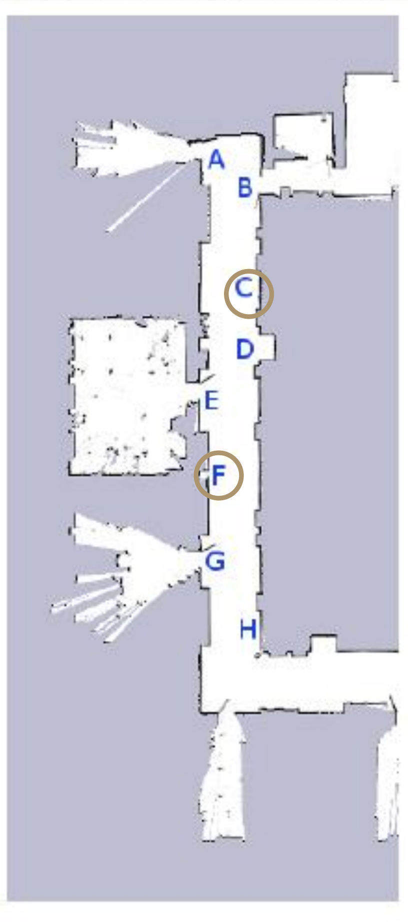
- 2 algorithms SEBS/DTAP

- **DTAP**
  higher avg
  lower stddev
  few outliers



Tests with robots





Tests with robots

Conclusions

Conclusions

Online DTA for practical approaches to MRP

- patrolling_sim available (standard benchmark for MRP)

  http://wiki.ros.org/patrolling_sim

Future Work

- Apply novel DTA approaches
  - improved robustness to comm. delays, failures
- Including learning techniques for adaptive behaviors