RECRUITMENT NEAR WORKSITES FACILITATES ROBUSTNESS OF FORAGING E-PUCK SWARMS TO GLOBAL POSITIONING NOISE

Lenka Pitonakova, Alan Winfield, Richard Crowder

UNIVERSITY OF Southampton

UWE BRISTOL University of the West of England

brl Bristol Robotics Laboratory

University of BRISTOL

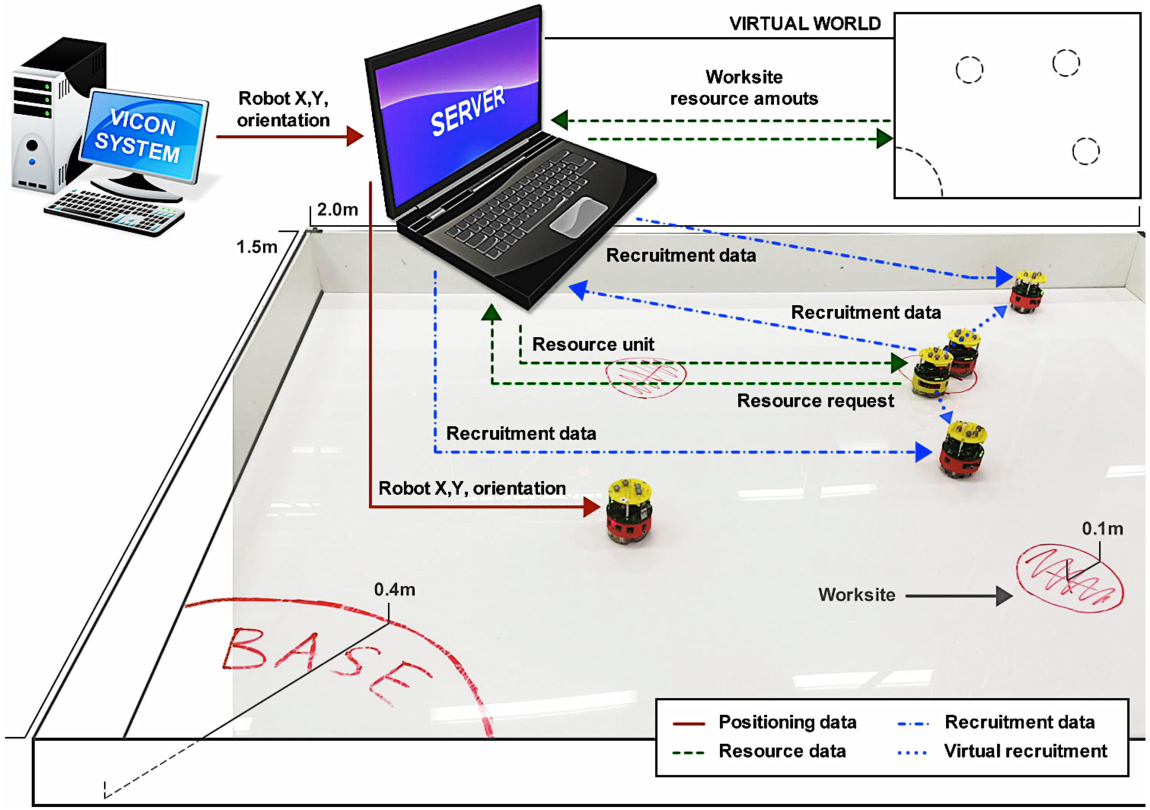
THALES

Overview

- □ Five e-pucks need to search the arena and find randomly-distributed worksites
- □ They then carry virtual resource units from worksites to the base until all worksites are depleted
- □ Do robot swarms where robots recruit each other perform better?
- □ How does noise in the GPS affect the swarms?

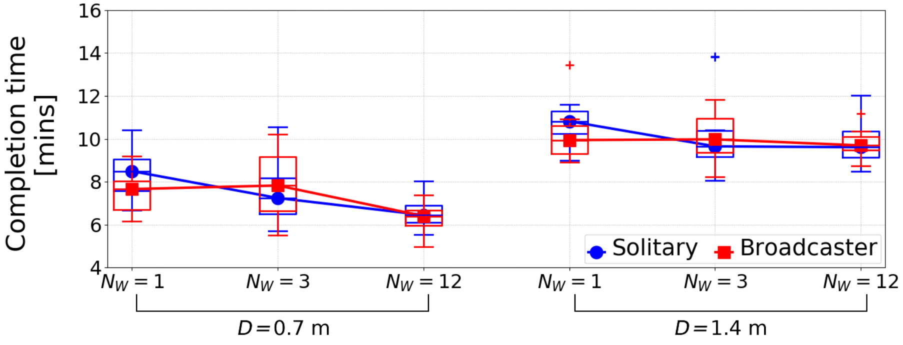
Semi-virtual environment

- Physical interactions
- Data handled by a server
  - Robot tracking and positioning
  - World state
  - Communication



Outcomes

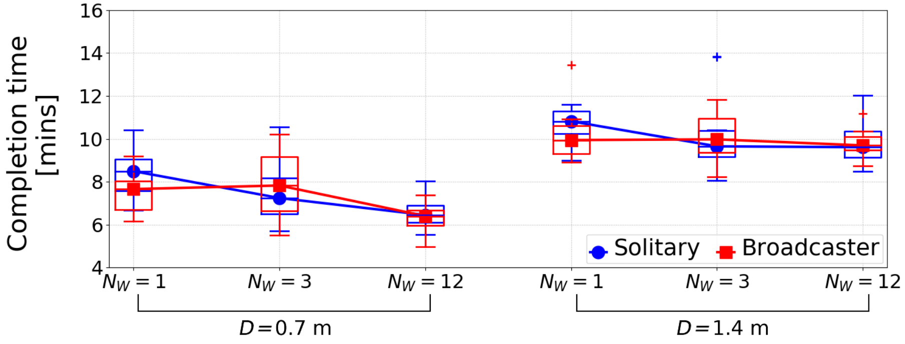
- □ In small noiseless environments, the choice of a foraging strategy does not matter
- □ When GPS noise is added, robots may loose track of foraging sites



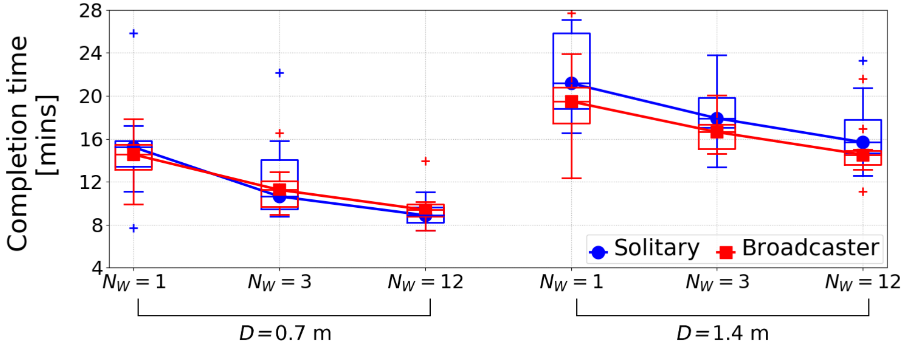
Without noise

Outcomes

- □ Recruitment near worksites helps the swarm maintain correct information about worksite locations
- □ Inherent sensory-motor noise makes obstacle avoidance harder in reality than in simulation



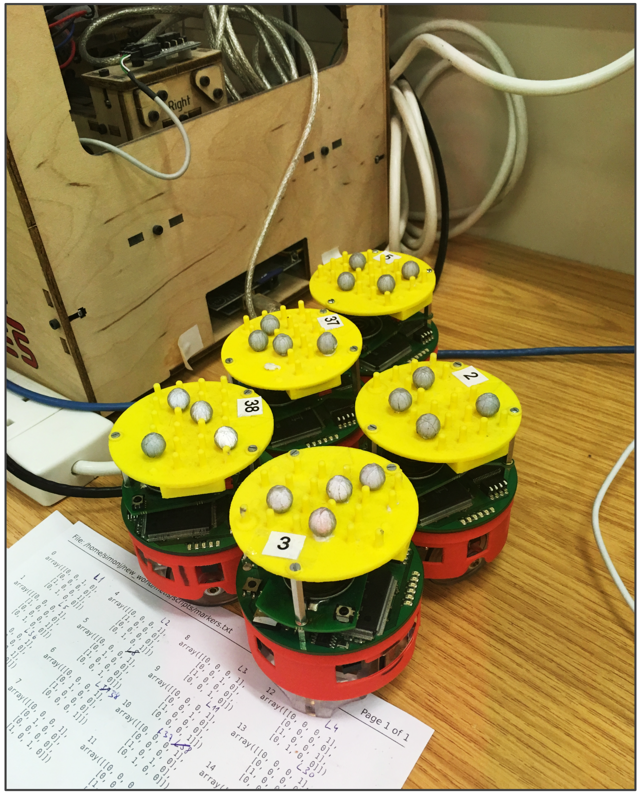
Without noise



With noise

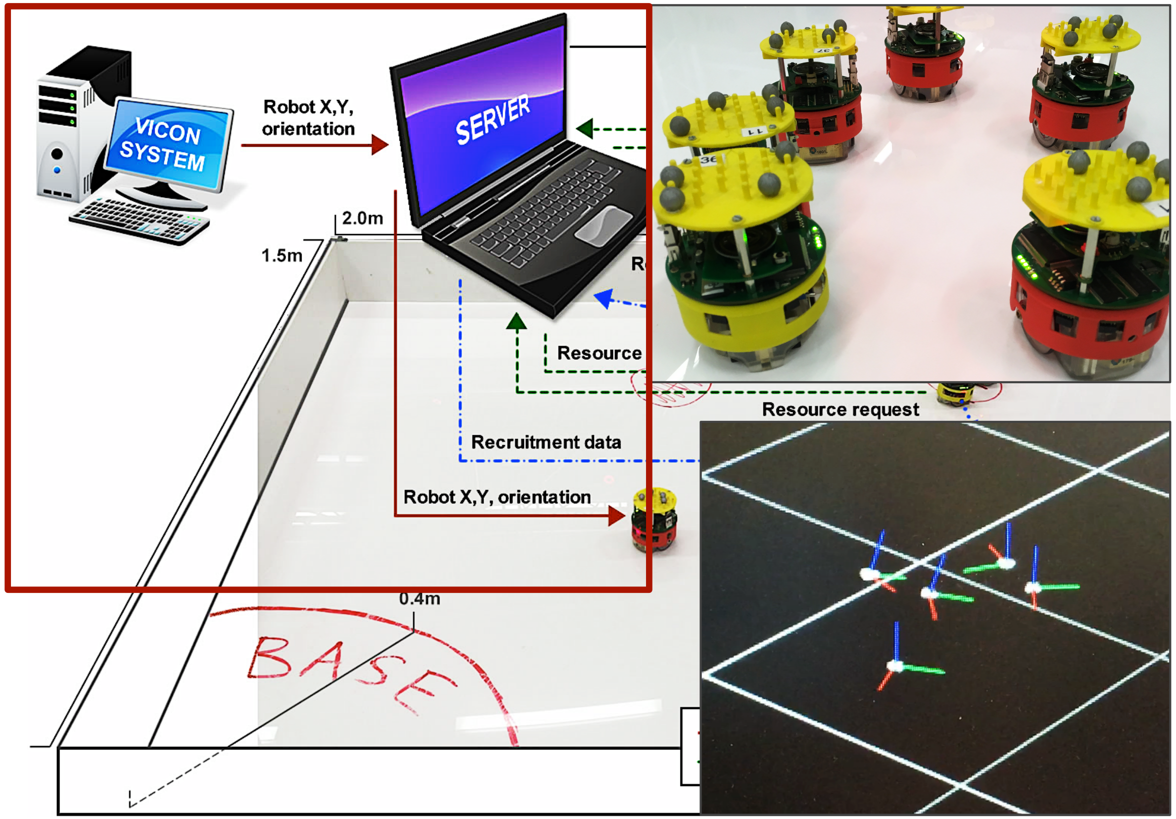
Experiment details

Parameter	Value
Number of robots	5
Arena size	2 × 1.5 m
Base radius	0.4 m
Worksite radius	0.1 m
Number of worksites	{1,3,12}
Min. worksite distance from base edge	{0.7,1.4} m
Total reward	48
Robot worksite sensor range	0.25 m
Robot communication range	1.25 m



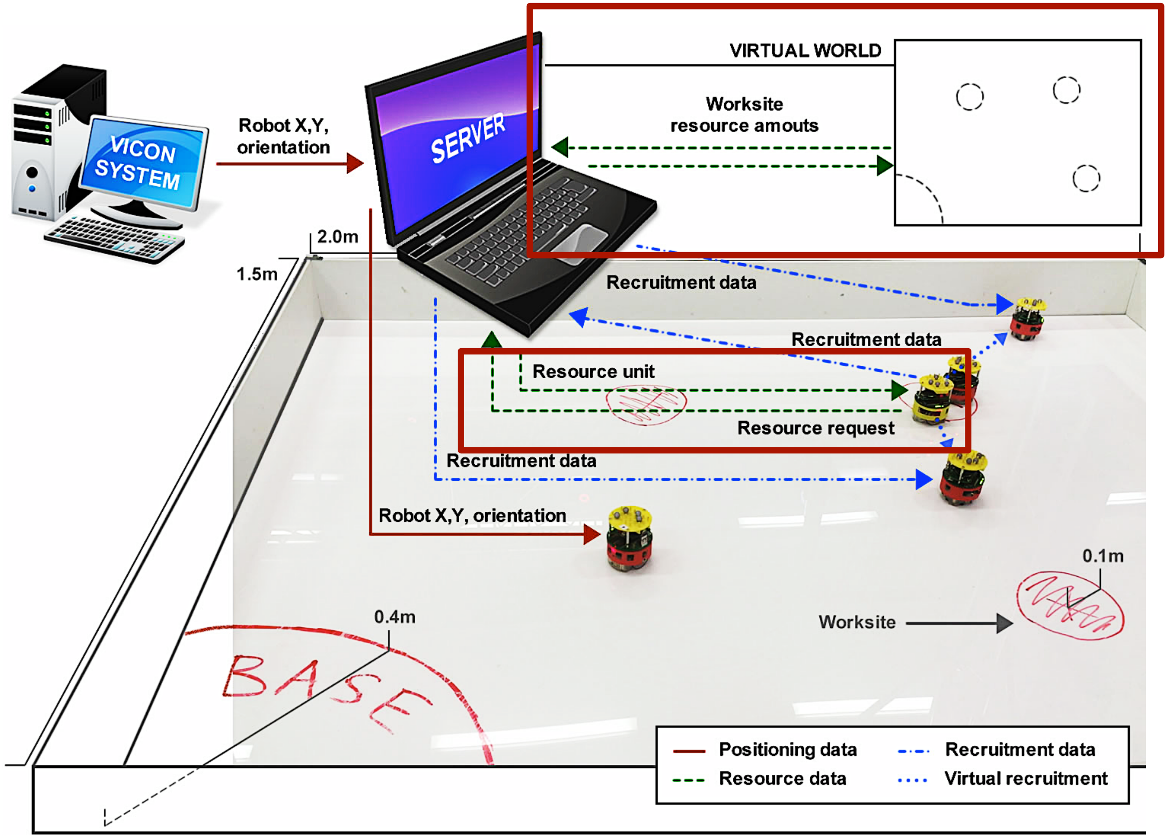
Semi-virtual environment: Positioning

- □ Vicon tracking system connected to the Server
- □ Server sends X,Y position to each robot every 1/10 seconds



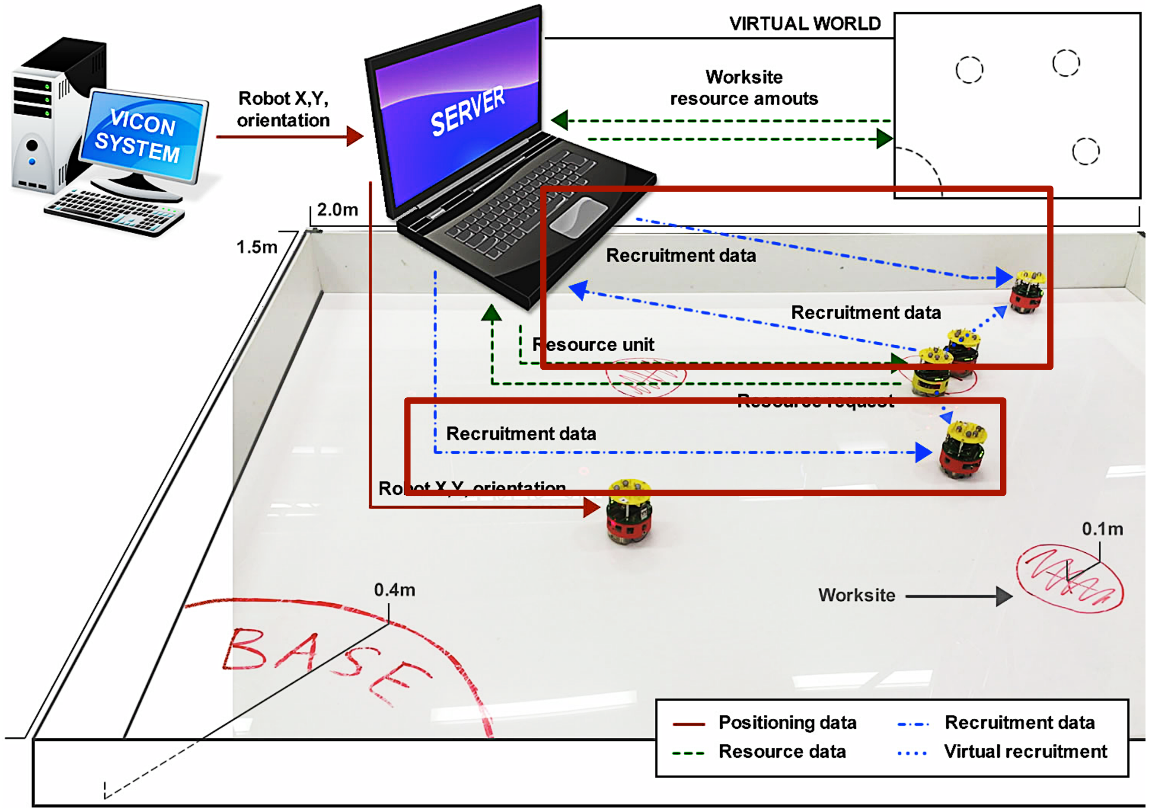
Semi-virtual environment: World state

- The Server knows worksite locations and resource left
- Robots that think they are at worksites request resource units from the Server



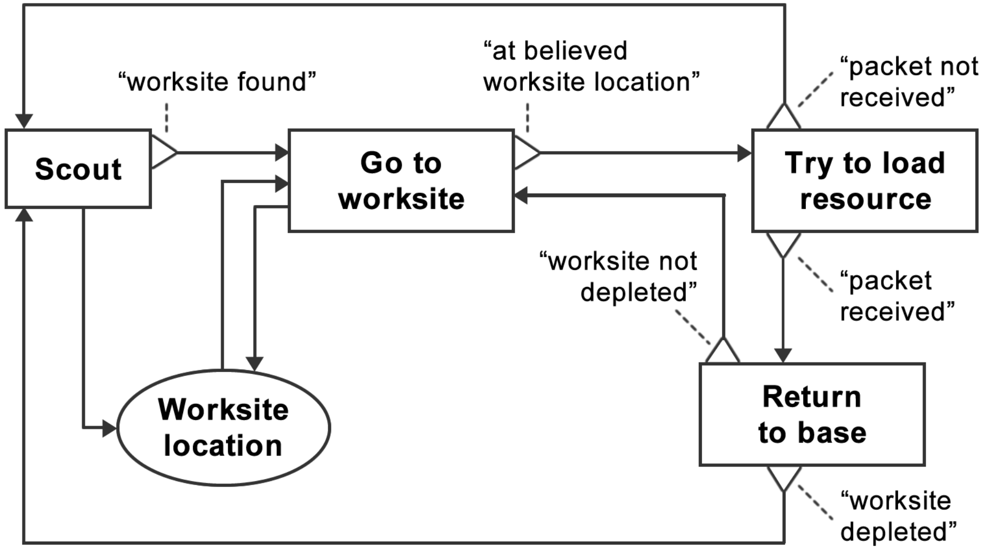
Semi-virtual environment: Recruitment

- □ Recruiters send recruitment signals and worksite location to the Server
- □ The Server sends the data to robots near the recruiter



Robot control algorithms: Solitary

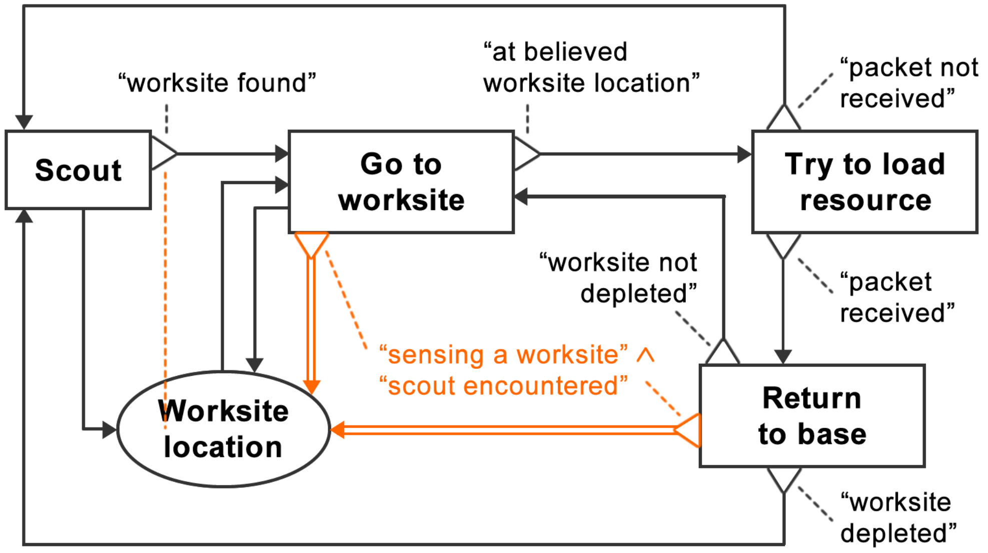
- □ Robots start in the base and scout for worksites
- □ When a robot finds a worksite, it tries to load resource from it
- □ The robot delivers resources to the base until the worksite is depleted or not found



BDRML representation of the Solitary controller

Robot control algorithms: Broadcaster

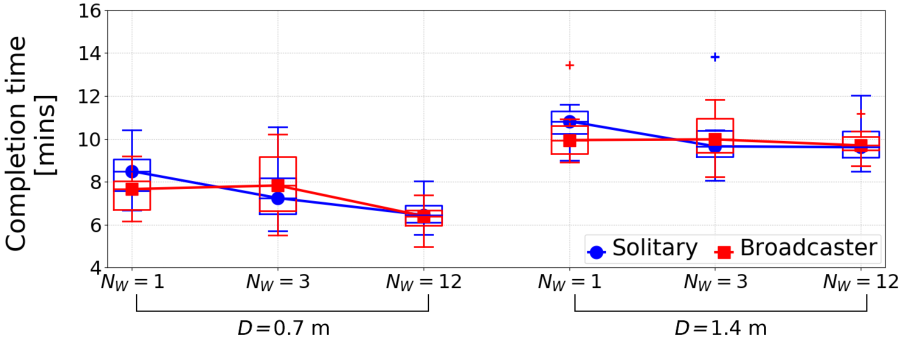
- □ Based on the Solitary controller, but robots also recruit each other
- □ While a robot is near a worksite, it sends recruitment signals with believed worksite location to nearby scouts



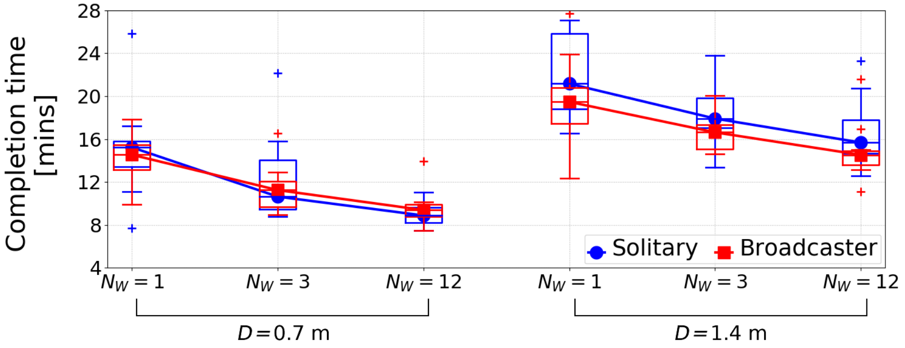
BDRML representation of the Broadcaster controller

Foraging performance

- □ Environments with different number of worksites (N_W) and worksite distance from the base (D)
- □ When there is no noise, both controllers perform similarly
- □ Noise increases completion time. The Solitary swarm is more affected.



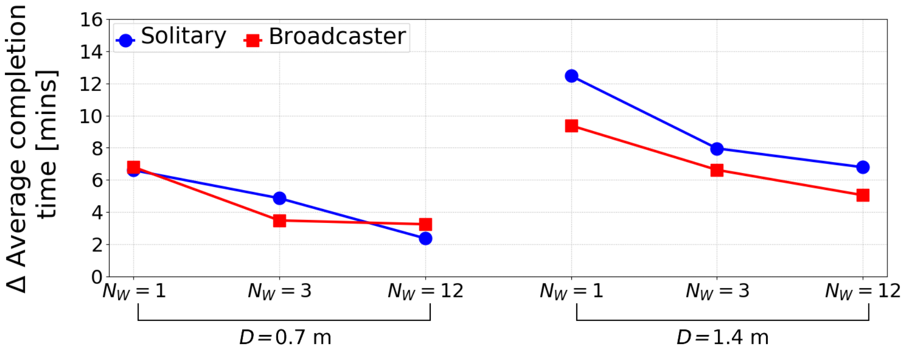
Without noise

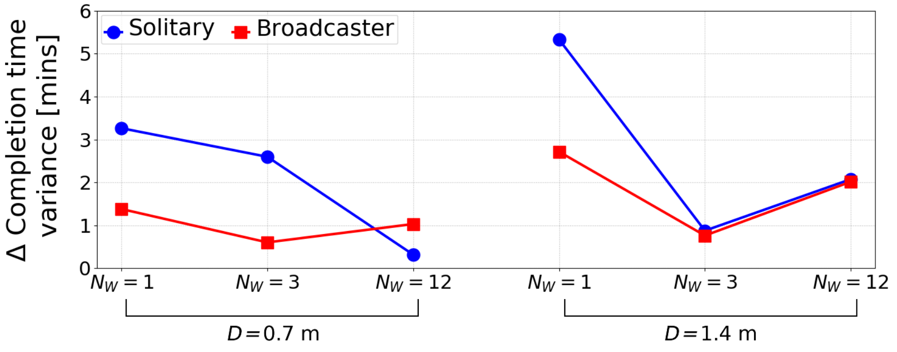


With noise

Foraging performance

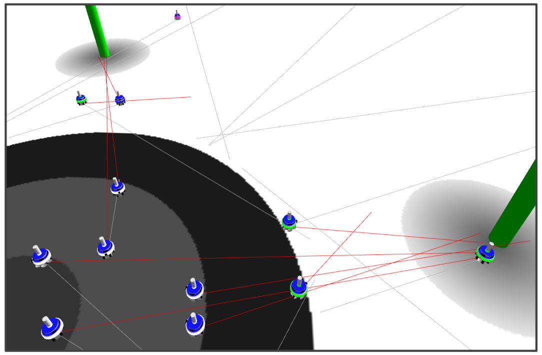
- As a result of noise, Solitary swarm exhibits a larger increase in:
  - Average completion time when *D* is large
  - Completion time variance when N_W is small

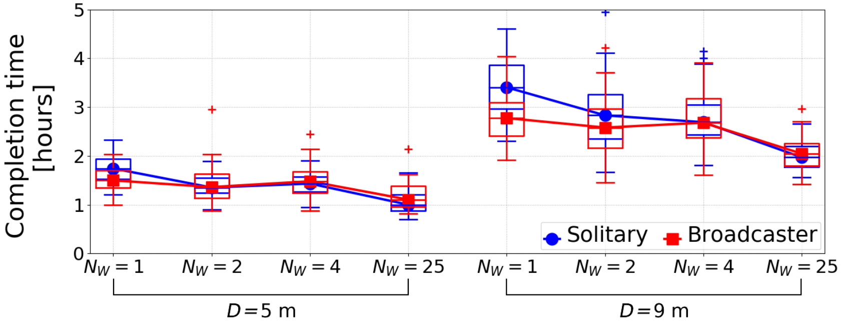




Relation to previous work

- Similar results than in larger simulated environments
- Obstacle avoidance and dealing with congestion are more difficult in real world due to inherent noise
- Controller type affects what kind of noise a swarm can be robust to





Task completion time in a larger simulation

RECRUITMENT NEAR WORKSITES FACILITATES ROBUSTNESS OF FORAGING E-PUCK SWARMS TO GLOBAL POSITIONING NOISE

Find out more:

Designing Robot Swarms project:
https://rebrand.ly/designingSwarms

The BDRML language
https://rebrand.ly/bdrml

Lenka Pitonakova, Alan Winfield, Richard Crowder

Acknowledgements

This work was supported by EPSRC grants EP/G03690X/1, EP/N509747/1 and EP/R004757/1 and by Thales Group. A video accompanying this paper is available on the IROS 2018 proceedings website. The source code and data are available on https://doi.org/10.5258/SOTON/D0388.

We would like to thank Wenguo Liu for providing his robot and computer programs that served as a basis for the programs used here.