CO-OPERATIVE ROBOTS SHARING THE LOAD

NATALIA CHECHINA

NCHECHINA@BOURNEMOUTH.AC.UK

@NCHECHINA

8 NOVEMBER 2018

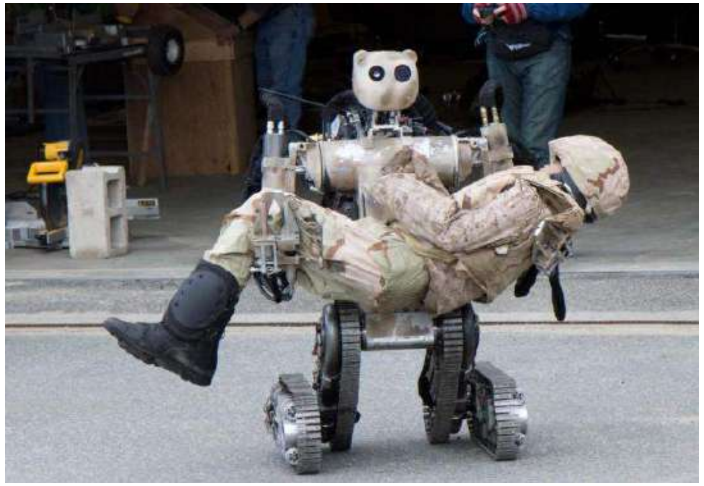
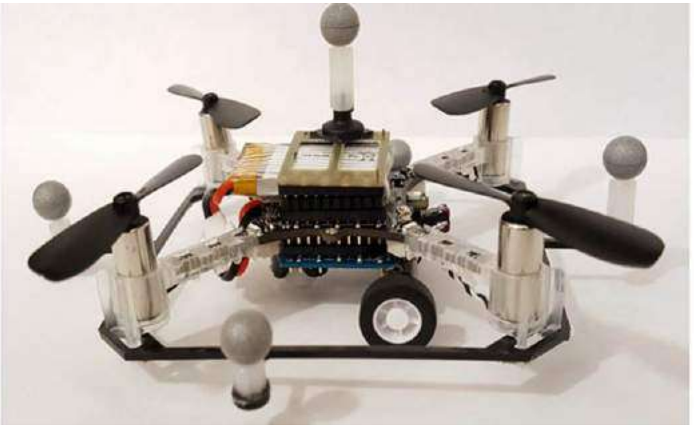
WHO AM I…

- Lecturer in Computing at Bournemouth University, UK
- Research interest – **Scaling distributed systems reliably**
- Students and colleagues from Heriot-Watt and Glasgow Universities

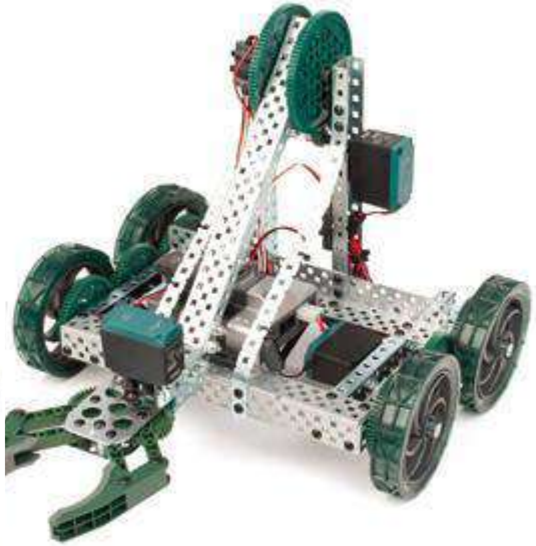








LOAD DISTRIBUTION!

LOAD DISTRIBUTION!

First law of distributed systems – DON'T

THINGS TO CONSIDER

- Decision making
  - Who collects state information?
  - Who decides where to move?
  - What is moved?

THINGS TO CONSIDER

- Decision making
  - Who collects state information?
  - Who decides where to move?
  - What is moved?
- Aim (obvious but often forgotten concept)
  - Even load distribution?
  - Even process/program/item distribution?
  - Reducing program completion time?

THINGS TO CONSIDER

- Decision making
  - Who collects state information?
  - Who decides where to move?
  - What is moved?
- Aim
  - Even load distribution?
  - Even process/program/item distribution?
  - Reducing program completion time?
- Complexity of the algorithm
  - To work at scale it must be simple

LOAD DISTRIBUTION FOR ROBOTS

Autonomous Mobile Programs (AMPs)

@NCHECHINA

AUTONOMOUS MOBILE PROGRAMS

- Decision making
  - The whole program moves
  - Each AMP decides when and where to move
  - Information is collected by so called “Load Server” → Blackboard

AUTONOMOUS MOBILE PROGRAMS

- Decision making
  - The whole program moves
  - Each AMP decides when and where to move
  - Information is collected by so called “Load Server” → Blackboard
- Aim
  - Selfish
  - Each AMP aims to reduce its own completion time

AUTONOMOUS MOBILE PROGRAMS

- Decision making
  - The whole program moves
  - Each AMP decides when and where to move
  - Information is collected by so called “Load Server” → Blackboard
- Aim
  - Selfish
  - Each AMP aims to reduce its own completion time
- Complexity
  - Simple cost model

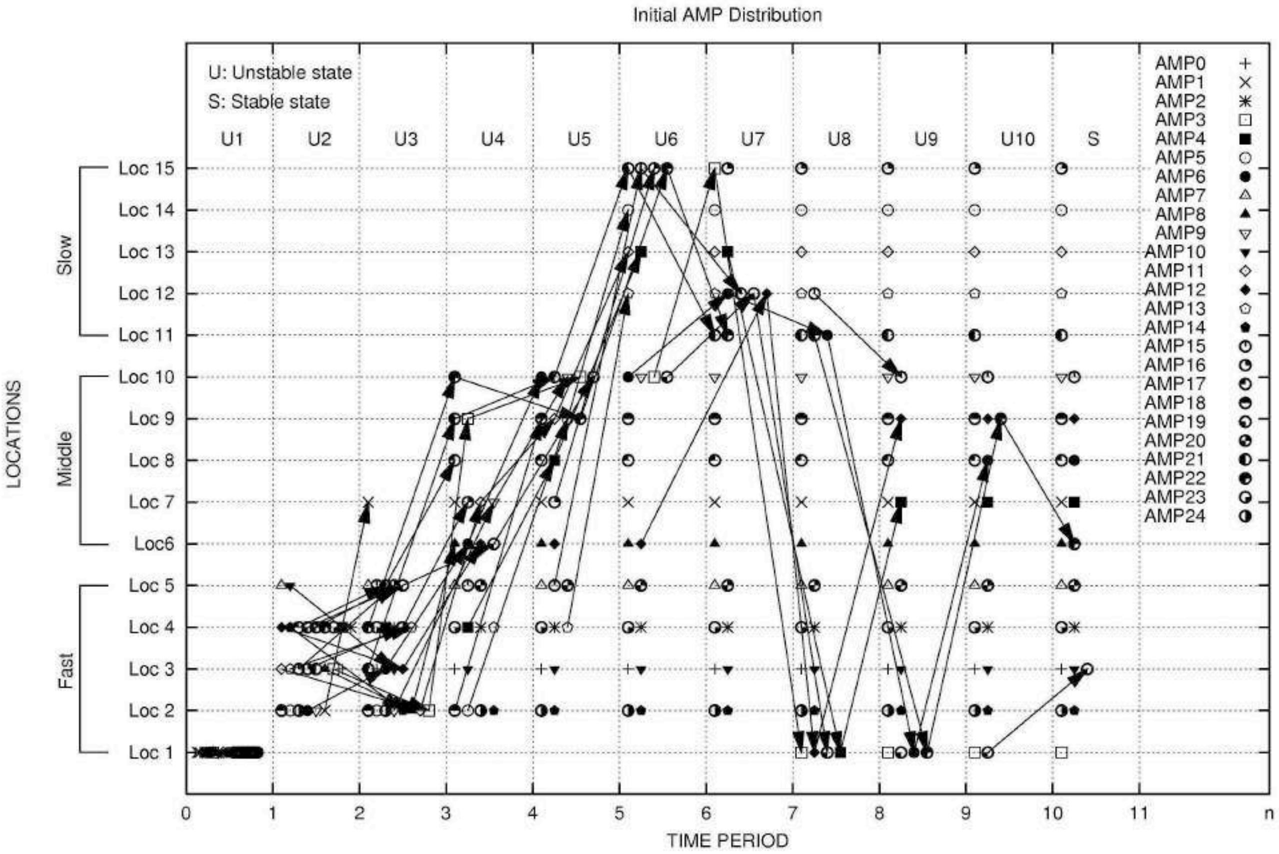
$$T_{here} > T_{there} + T_{comm}$$

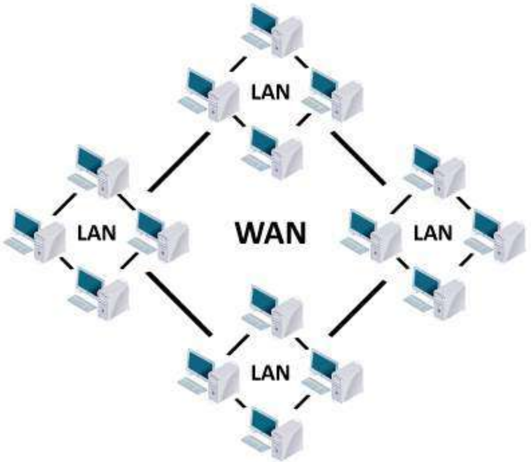



INITIAL RESEARCH

@NCHECHINA

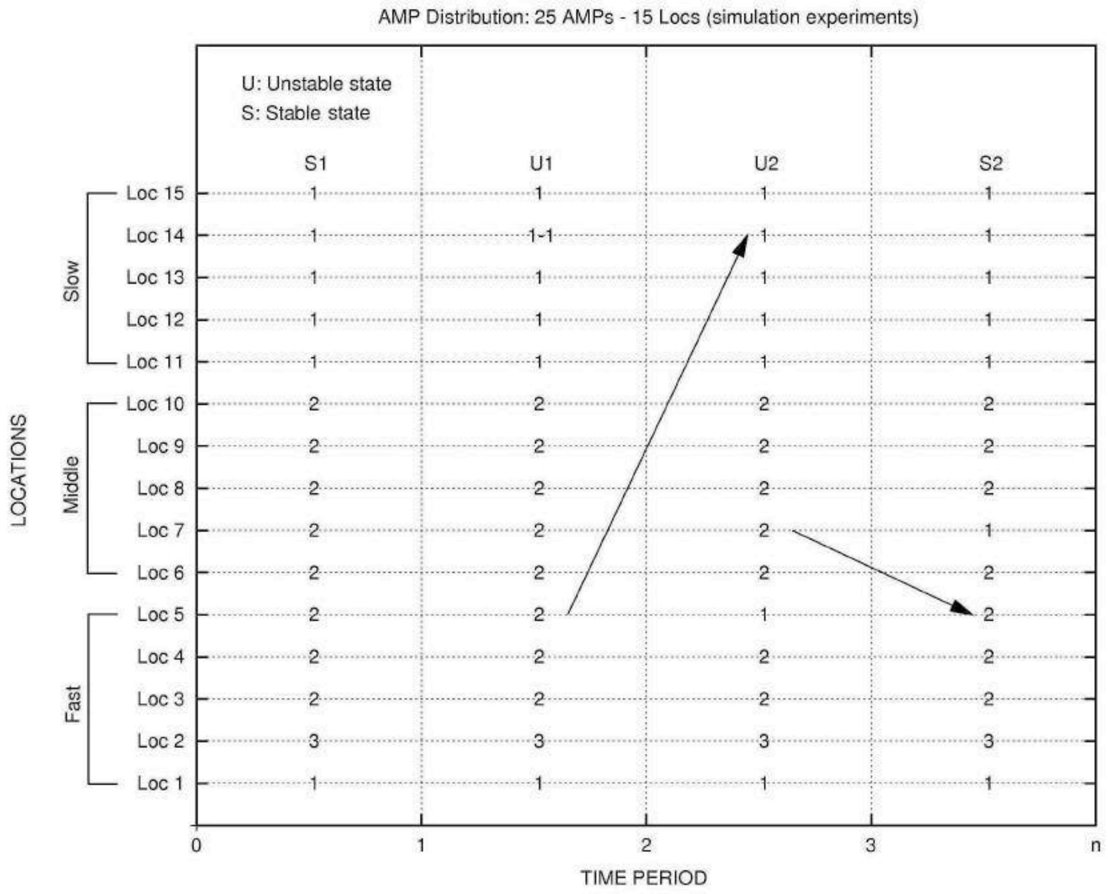
AUTONOMOUS MOBILE PROGRAMS (AMPs)

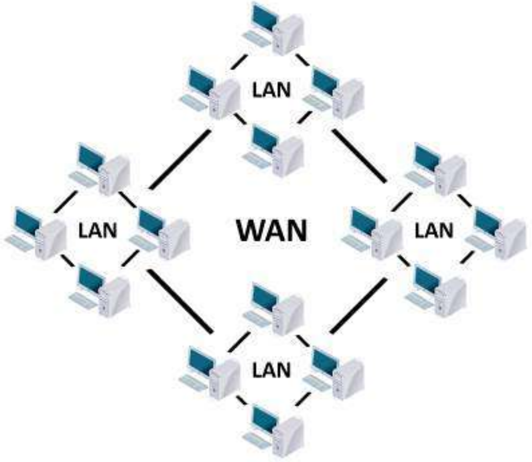




N. Chechina, P. King, and P. Trinder. *Redundant Movements in Autonomous Mobility: Experimental and Theoretical Analysis*. Journal of Parallel and Distributed Computing (JPDC), Elsevier, Volume 71, Issue 10, October 2011, pp. 1278--1292.

AUTONOMOUS MOBILE PROGRAMS (AMPs)





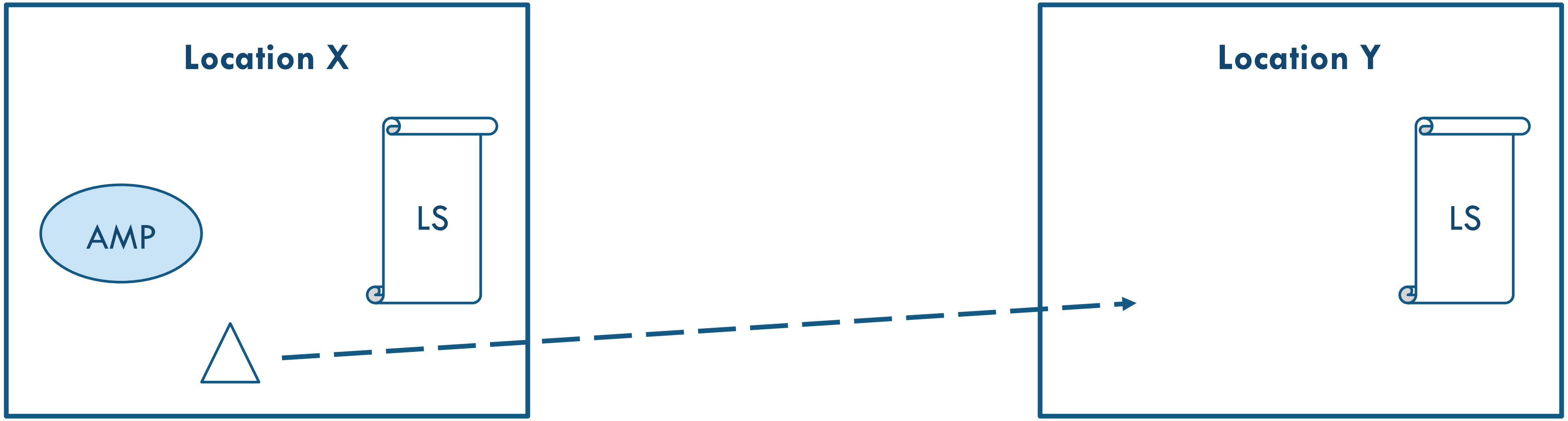
N. Chechina, P. King, and P. Trinder. *Redundant Movements in Autonomous Mobility: Experimental and Theoretical Analysis*. Journal of Parallel and Distributed Computing (JPDC), Elsevier, Volume 71, Issue 10, October 2011, pp. 1278--1292.

IMPLEMENTATION

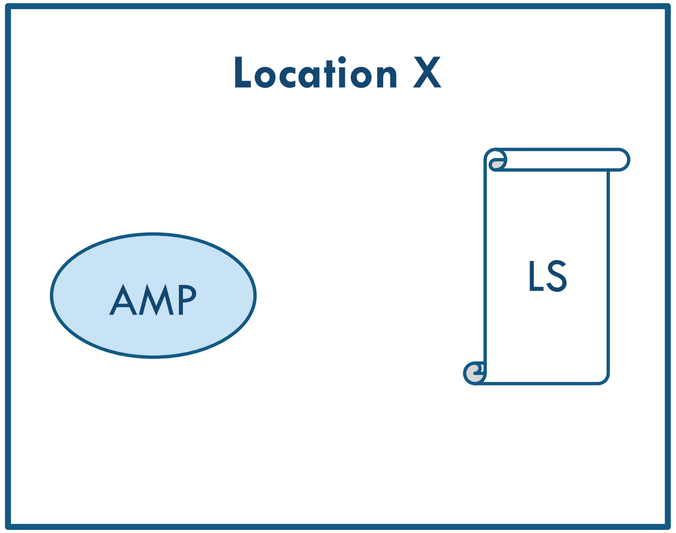
IMPLEMENTATION

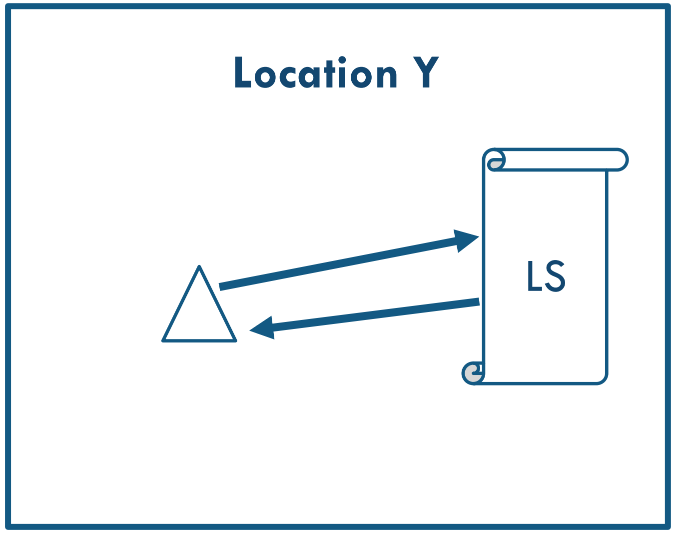
IMPLEMENTATION

IMPLEMENTATION

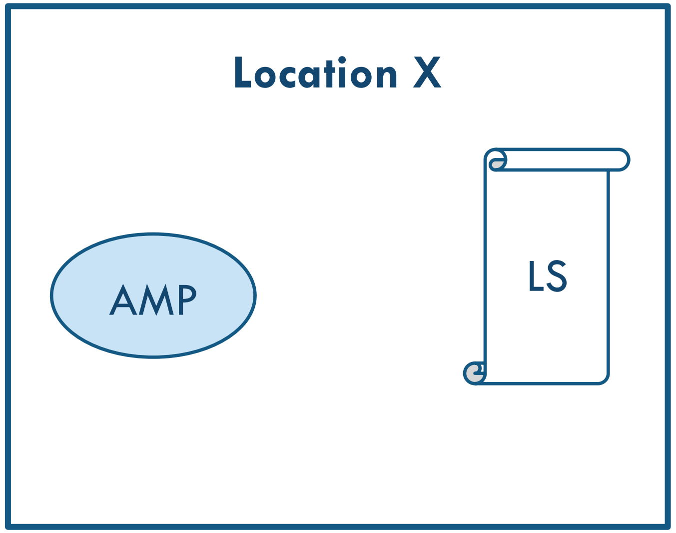


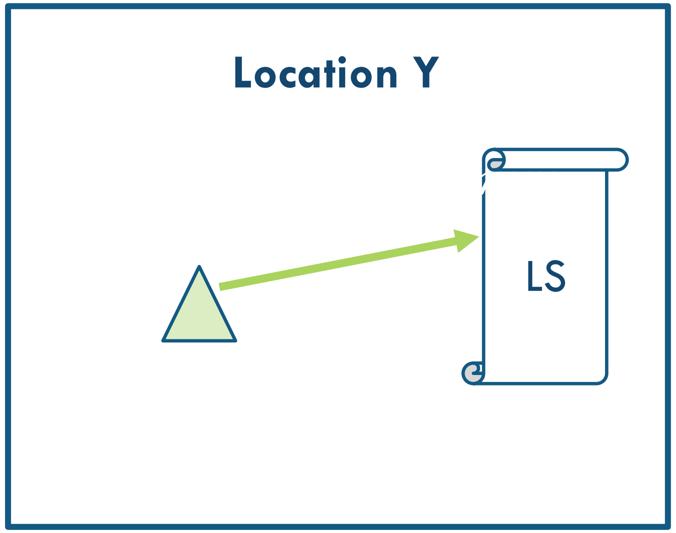
IMPLEMENTATION



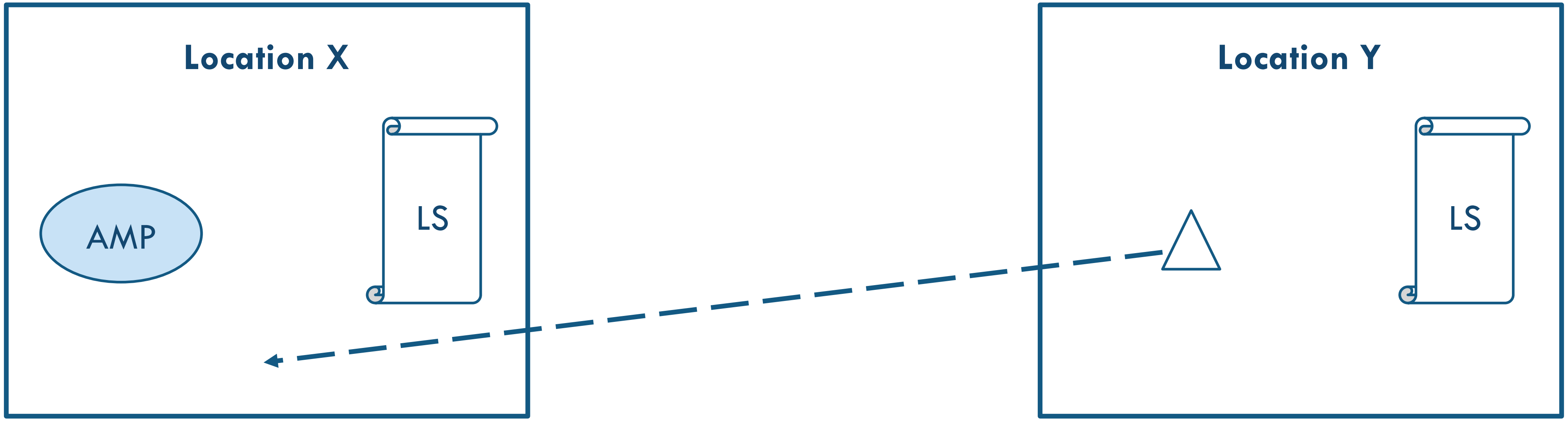


IMPLEMENTATION





IMPLEMENTATION

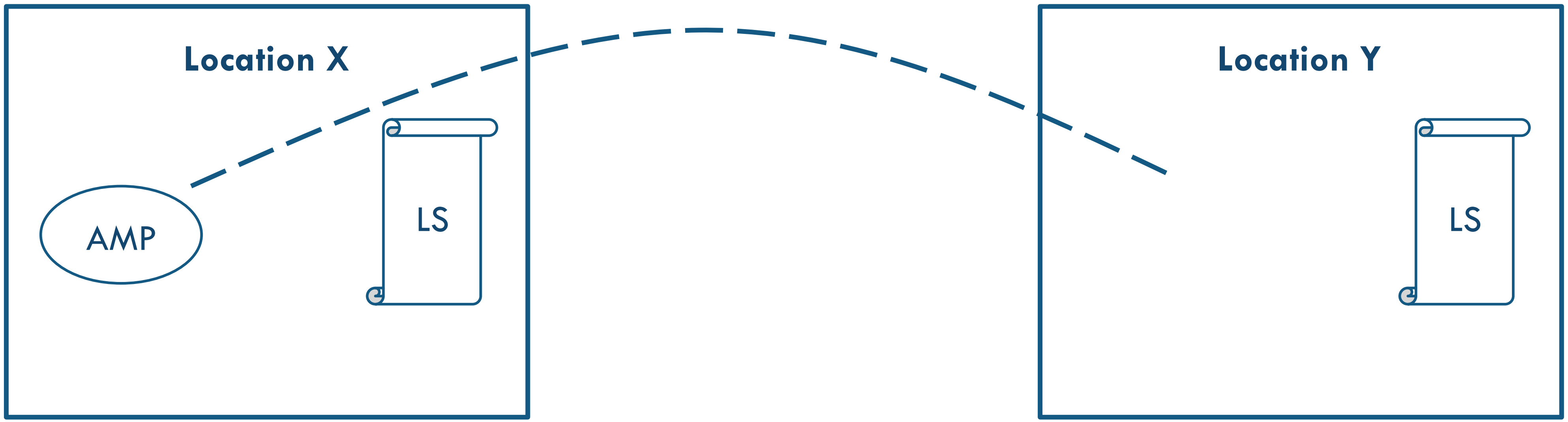


IMPLEMENTATION

IMPLEMENTATION

IMPLEMENTATION

IMPLEMENTATION



IMPLEMENTATION

IMPLEMENTATION

IMPLEMENTATION

IMPLEMENTATION

EXPERIMENTS

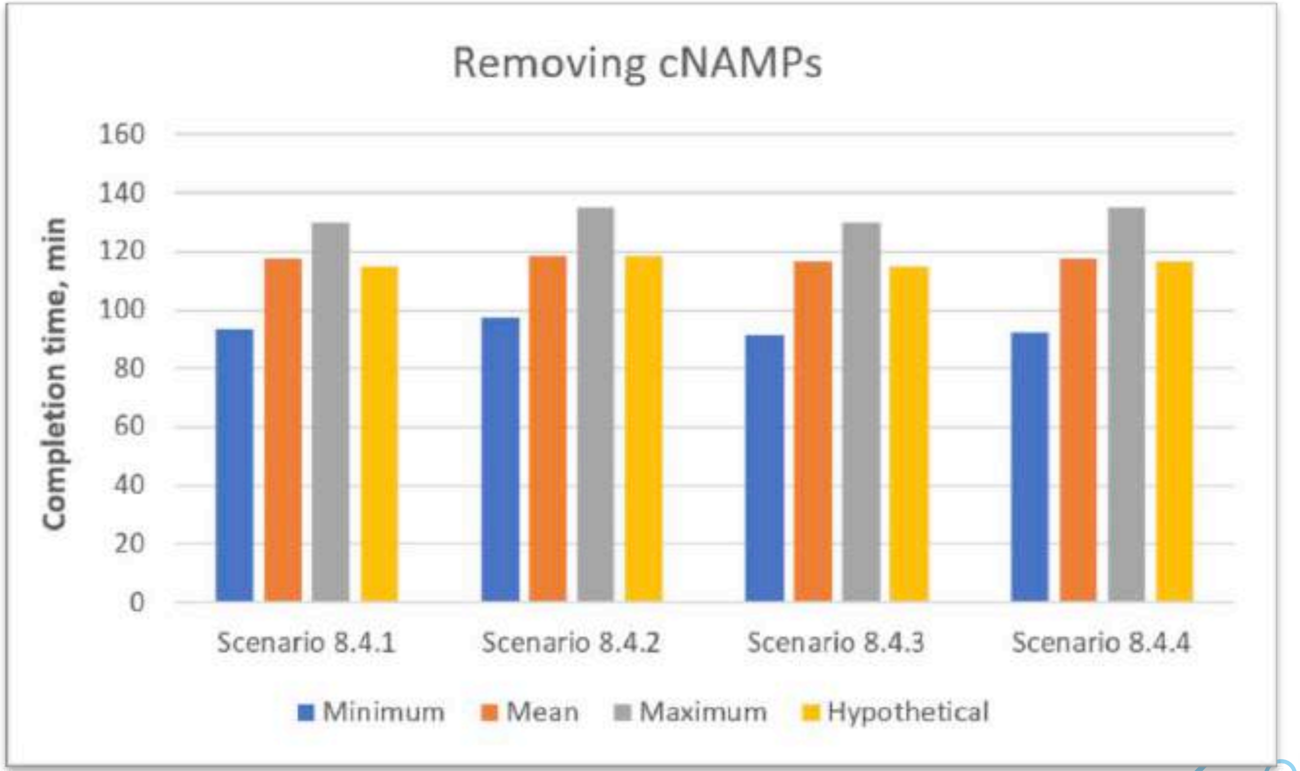
- Properties of balanced states
- (relatively) Large scale simulation
  - ~350 location, ~3500 AMPs
- Worst case analysis of redundant movements
- Maximum number, and probability of, redundant movements

- Networks
  - topologies
  - number of locations
  - speed of locations
- AMPs
  - number of AMPs
  - type of AMPs
- Rebalancing
  - initial distribution
  - rebalancing after adding AMPs
  - rebalancing after termination of AMPs

GREEDY EFFECTS

- Worst case (maximum number) of redundant movements
  - q subnetworks → at most (q − 1) redundant movements
  - T_{comm} , chunk execution, and “confirm before move” help a lot

  $T_{here} > T_{there} + T_{comm}$
- While some AMPs move, the remaining AMPs take advantage

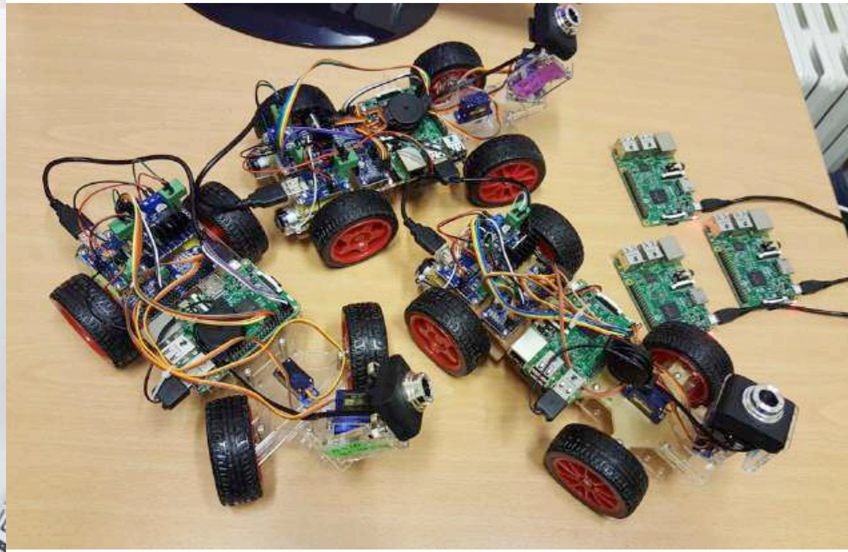





MOVING ON TO ROBOTS

@NCHECHINA

CAR KIT ROBOTS



http://www.dcs.gla.ac.uk/research/rosie/blog.html

ROBOT OPERATING SYSTEM

- Since 2007 simplifies the creation of complex robot behaviour across a wide variety of robotic platforms
- A de facto standard collection of
  - Tools
  - Libraries
  - Conventions

ROS

Open Source Robotics Foundation

NO SINGLE INDIVIDUAL, LABORATORY, OR INSTITUTION CAN HOPE TO DO IT ON THEIR OWN

PURPOSE

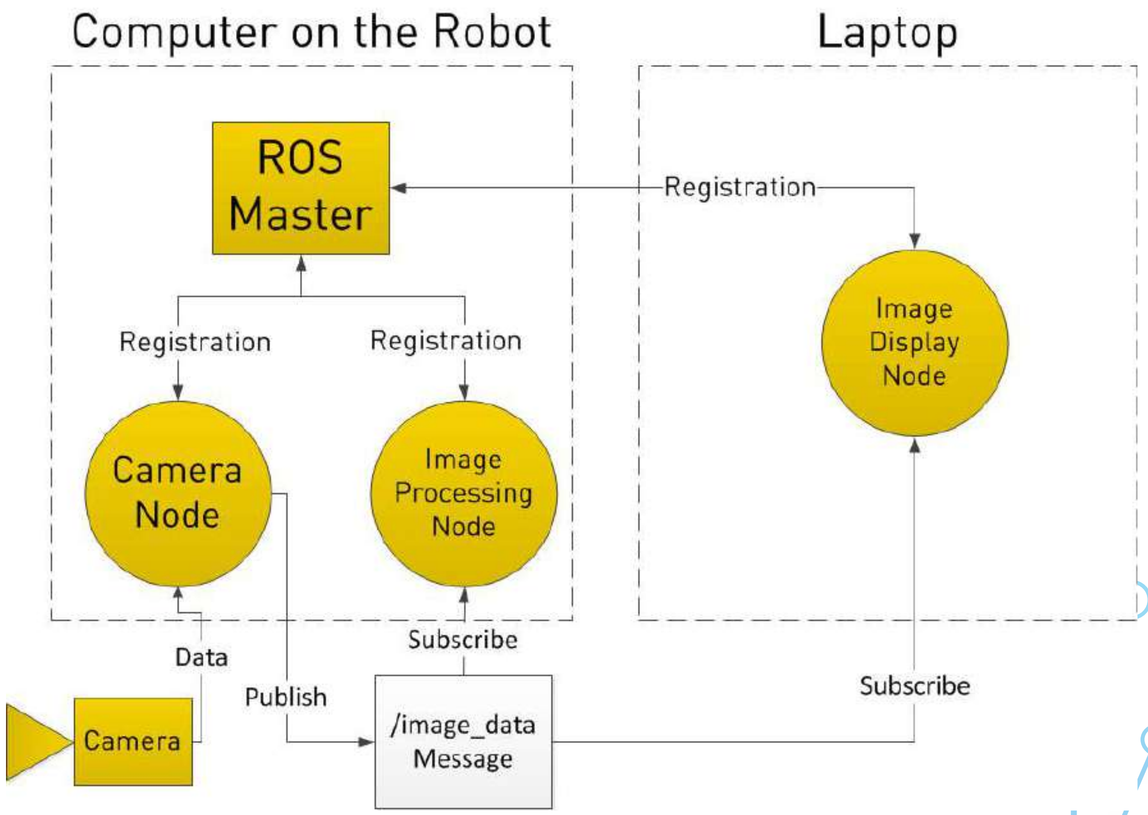
- Access to
  - Hardware drivers
  - Generic robot capabilities
  - Development tools
  - External libraries
  - etc...
- Systems may use as much or as little of ROS
- Encourage collaborative development of robotics software

SOME ROS STATISTICS (JULY 2018)

- >1900 people of active community
- >10 million lines of code
- ~4800 research papers acknowledge ROS
- ~130 types of robots support ROS

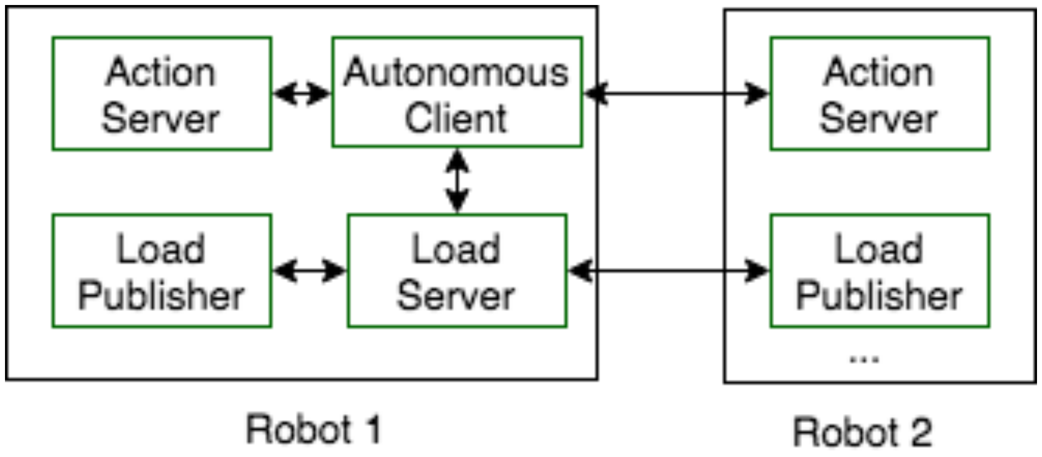
OVERVIEW

- Distributed Modular Design
- Shared development of common components
- Publish/subscribe message passing
  - Any node can subscribe to any other node
- Master node
  - Registration of all nodes



IMPLEMENTATION

- 5 robots
- Standard mains power
- All-to-all connection
- WiFi communication via a router
- ~~Programs~~ Tasks migrate



FAILURES TOLERATED IN RAMPS

	Failure	Tolerated in RAMP
1	Non-origin robot crashes due to any reason (hardware/software failure, power outage)	Yes
2	Non-origin robot loses network connection	Yes
3	Origin robot loses network connection	Yes

FAILURES TOLERATED IN RAMPS

	Failure	Tolerated in RAMP
1	Non-origin robot crashes due to any reason (hardware/software failure, power outage)	Yes
2	Non-origin robot loses network connection	Yes
3	Origin robot loses network connection	Yes
4	Origin robot crashes due to any reason (hardware/software failure, power outage)	No
5	Any sensors, motors, cameras, etc. fail without causing a robot crash	No

IMPLEMENTATION

- Task: route planning
  - MiniZinc and Gecode
- A task takes ~50s on a single core

IMPLEMENTATION

- Task: route planning
  - MiniZinc and Gecode
- A task takes ~50s on a single core
- RAMPs do not relocate
  - The initialising program stays on the initial robot
  - RAMP -- rout planning task
  - Weak mobility: RAMPs don’t carry state → after a move they restart computation

NETWORK AND ROBOT FAILURES

- Number of robots is fixed – 5
- Number of RAMPs fixed – 15

	Dist. \ Robot	R1	R2	R3	R4	R5
Run 1	D1	3	3	3	3	3
	D2	3	4	4	4	
	D3	4	6	5		
	D4	7	8			
	D5	15				

with time to relocate

Failed \ Robot	R1	R2	R3	R4	R5
none	2.67	3	3.25	3.08	3
R5	3.67	3.67	3.67	4	
R4,R5	5	5.33	4.67		
R3,R4,R5	7.33	7.67			
R2,R3,R4,R5	15				

no time to relocate

MIXED FAILURES

- Number of robots is fixed – 5
- Number of RAMPs fixed – 15

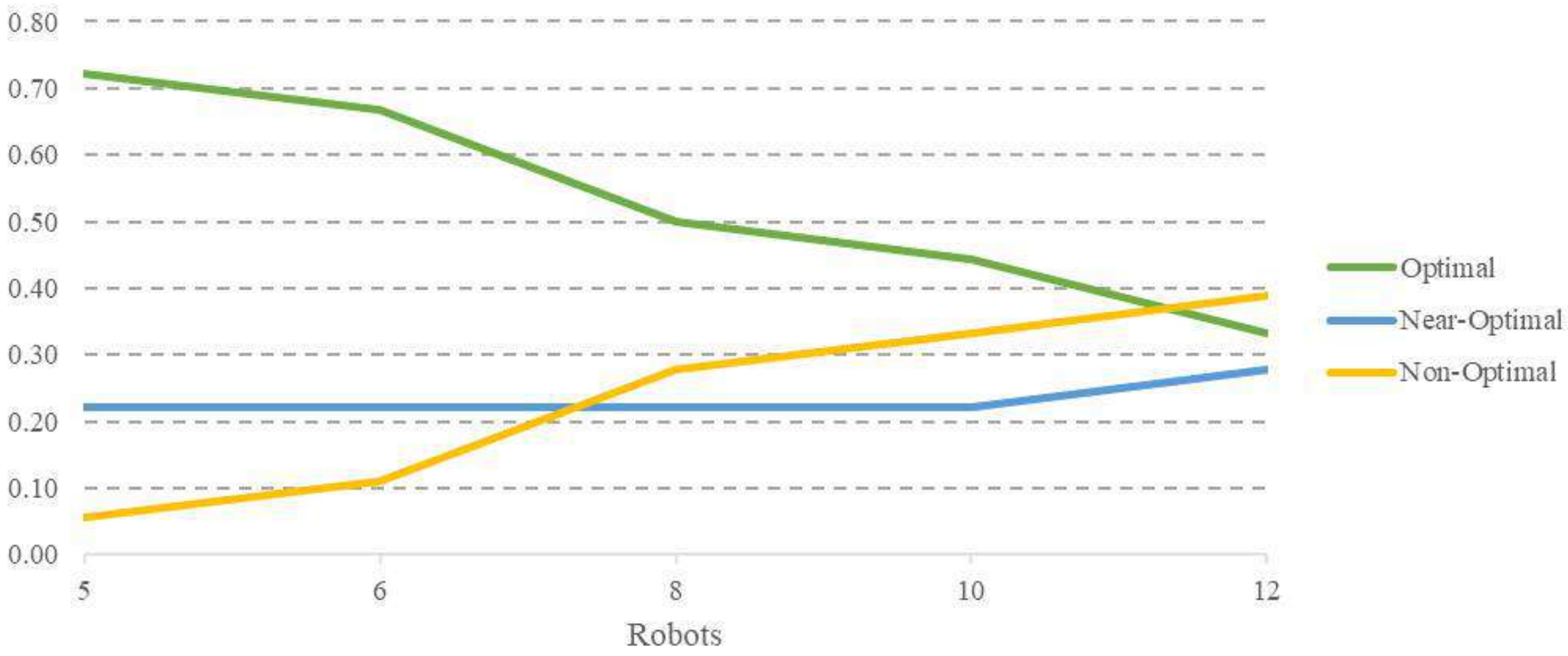
	Dist. \ Robot	R1	R2	R3	R4	R5
Run 1	D1	2	4	3	3	3
	D2	3	4	4	4	NF
	D3	6	5	4	RC	NF
	D4	7	8	NF	RC	NF
	D5	15	RC	NF	RC	NF

with time to relocate

Failed \ Robot	R1	R2	R3	R4	R5
none	3	2.92	2.92	3.25	2.92
R5	3.67	4	4	3.33	NF
R4,R5	4.33	5.67	5	RC	NF
R3,R4,R5	7.33	7.67	NF	RC	NF
R2,R3,R4,R5	15	RC	NF	RC	NF

no time to relocate

SCALING THE NUMBER OF ROBOTS



SCALING THE NUMBER OF RAMPs

- Number of robots is fixed – 5
- Number of RAMPs varies – 5—160

	Robots					
# of RAMPs	R1	R2	R3	R4	R5	
5	1	1	1	1	1	
10	2	2	2	2	2	
20	4	4	4	4	4	
40	8	8	8	9	7	2.5%
80	18	16	16	15	15	2.5%
160	31	30	35	32	32	1.8%

SUMMARY & FUTURE WORK

- Promising results ☺
- Strong connection between the number of AMPs and the number of robots

SUMMARY & FUTURE WORK

- Promising results ☺
- Strong connection between the number of AMPs and the number of robots
- Group robots?
  - Optimal size?
- External load
  - How to account for it?
  - How will it impact the cost model?

THANK YOU!

nchechina@bournemouth.ac.uk

@nchechina

https://staffprofiles.bournemouth.ac.uk/display/nchechina