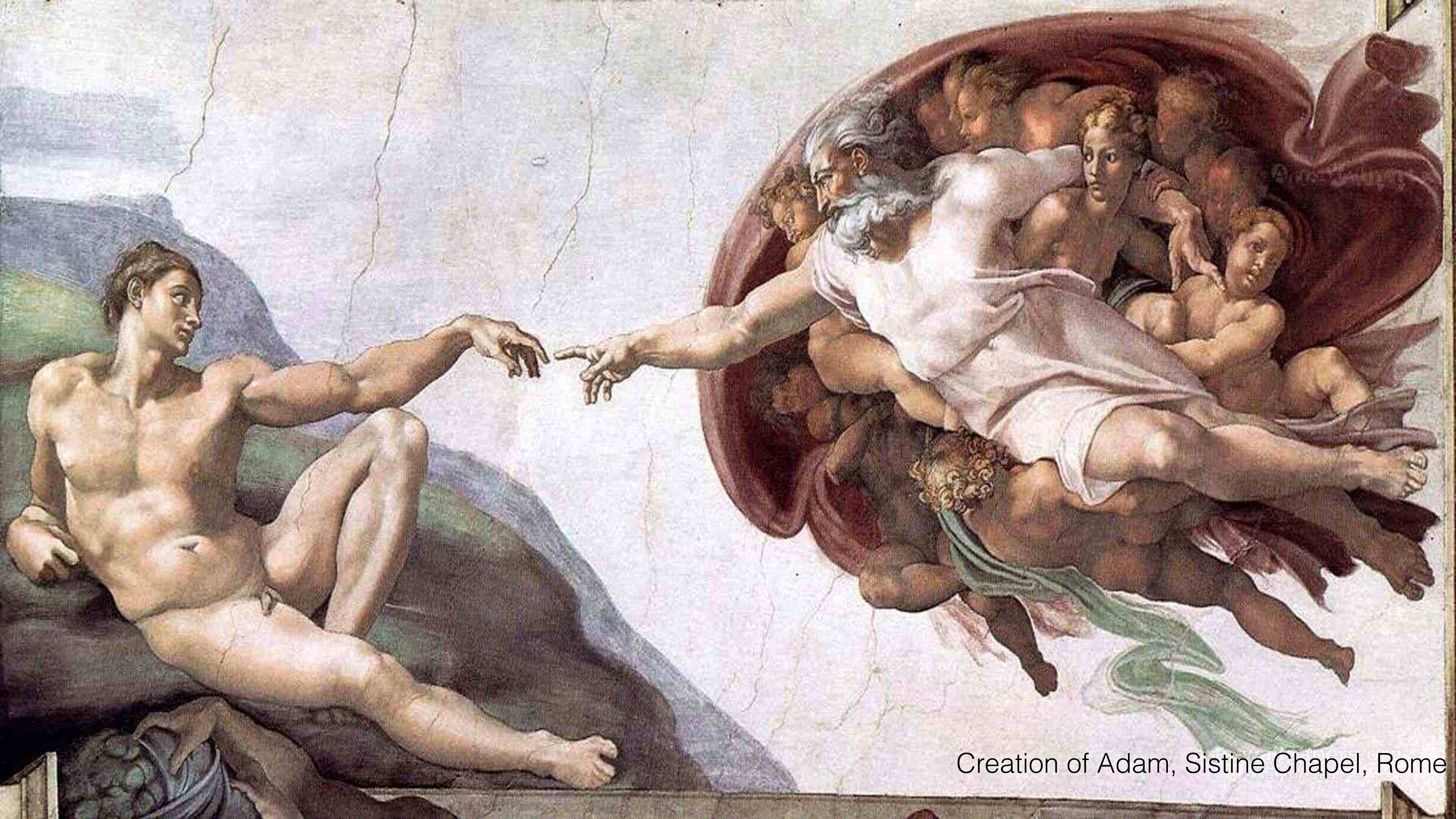


Few Rules to make Evolution an Intelligent Design Method

Vito Trianni

ISTC-CNR

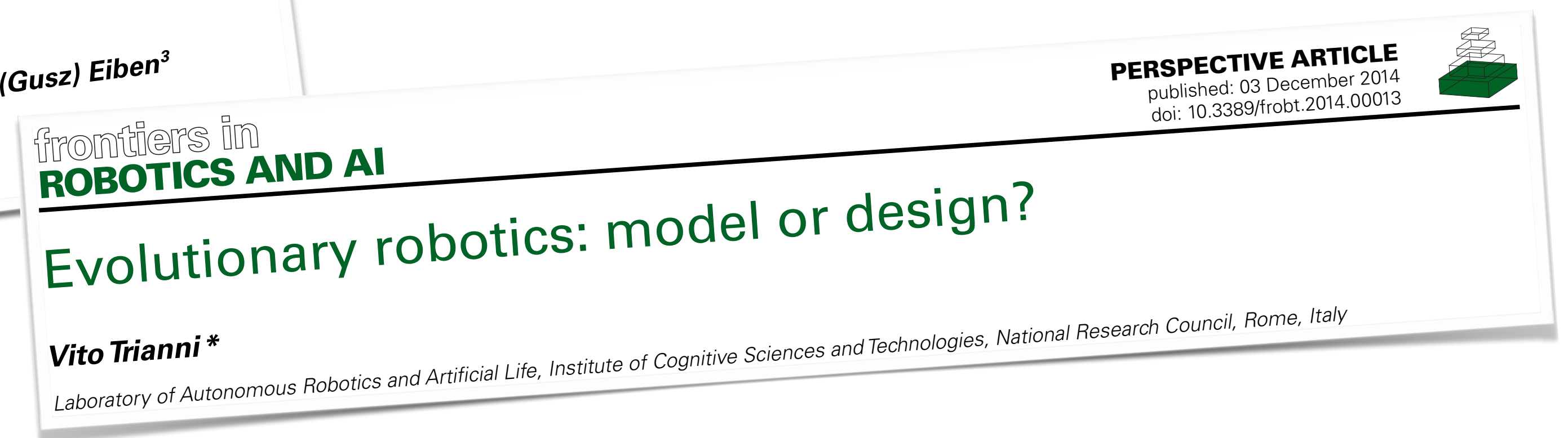
vito.trianni@istc.cnr.it



Creation of Adam, Sistine Chapel, Rome

ER: Evolution or Intelligent Design?

- The modeller:
“ER should be as close as possible to natural evolution!”
- The engineer:
“ER should be used as an (agnostic) method for design!”



ER as a design tool?

- ER is not a fully automatic design method
- Several **design choices** are needed to setup an ER experiment, which may introduce (spurious) selective pressures

Evol. Intel. (2014) 7:71–93
DOI 10.1007/s12065-014-0110-x

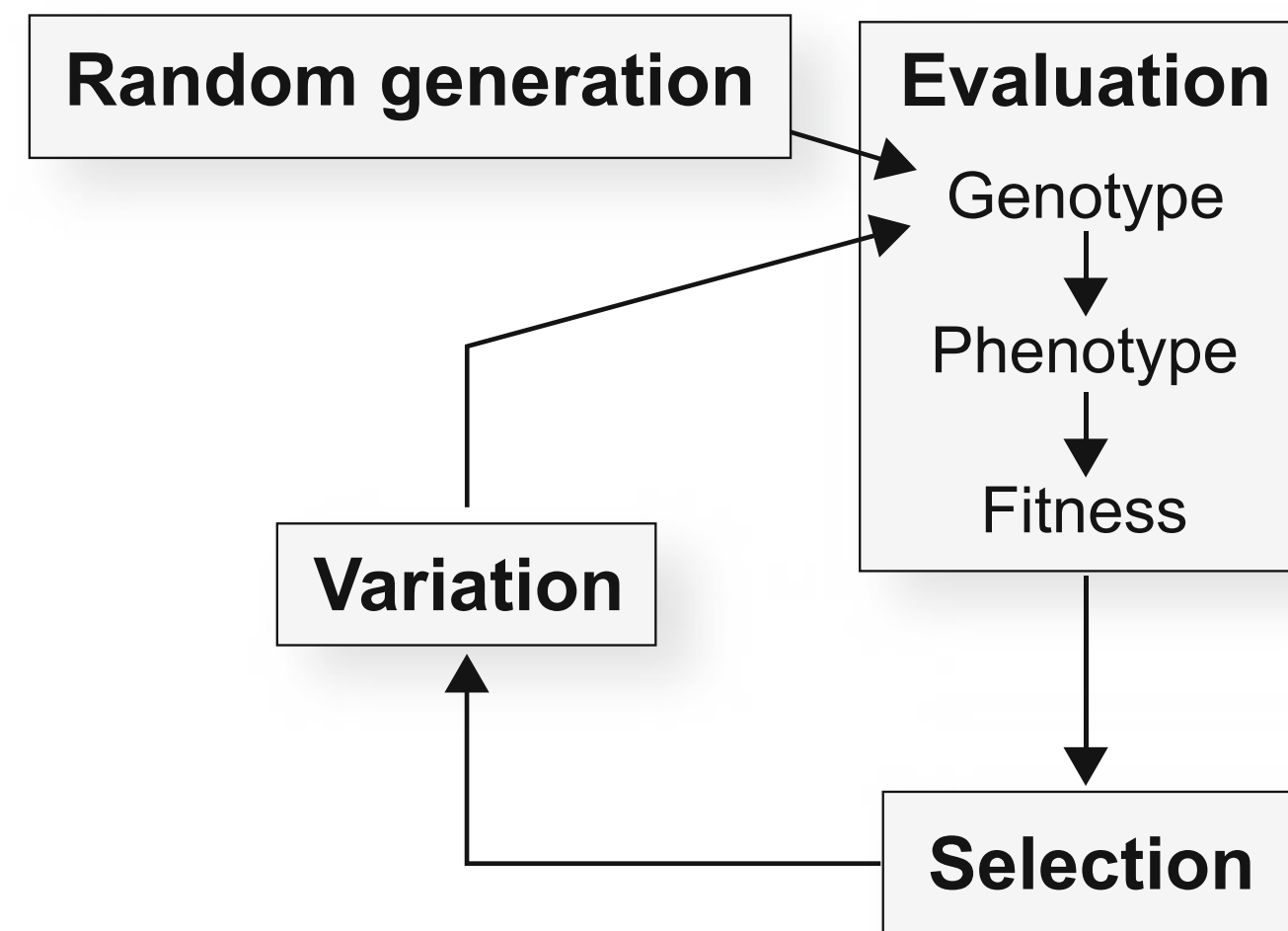
SPECIAL ISSUE

Beyond black-box optimization: a review of selective pressures for evolutionary robotics

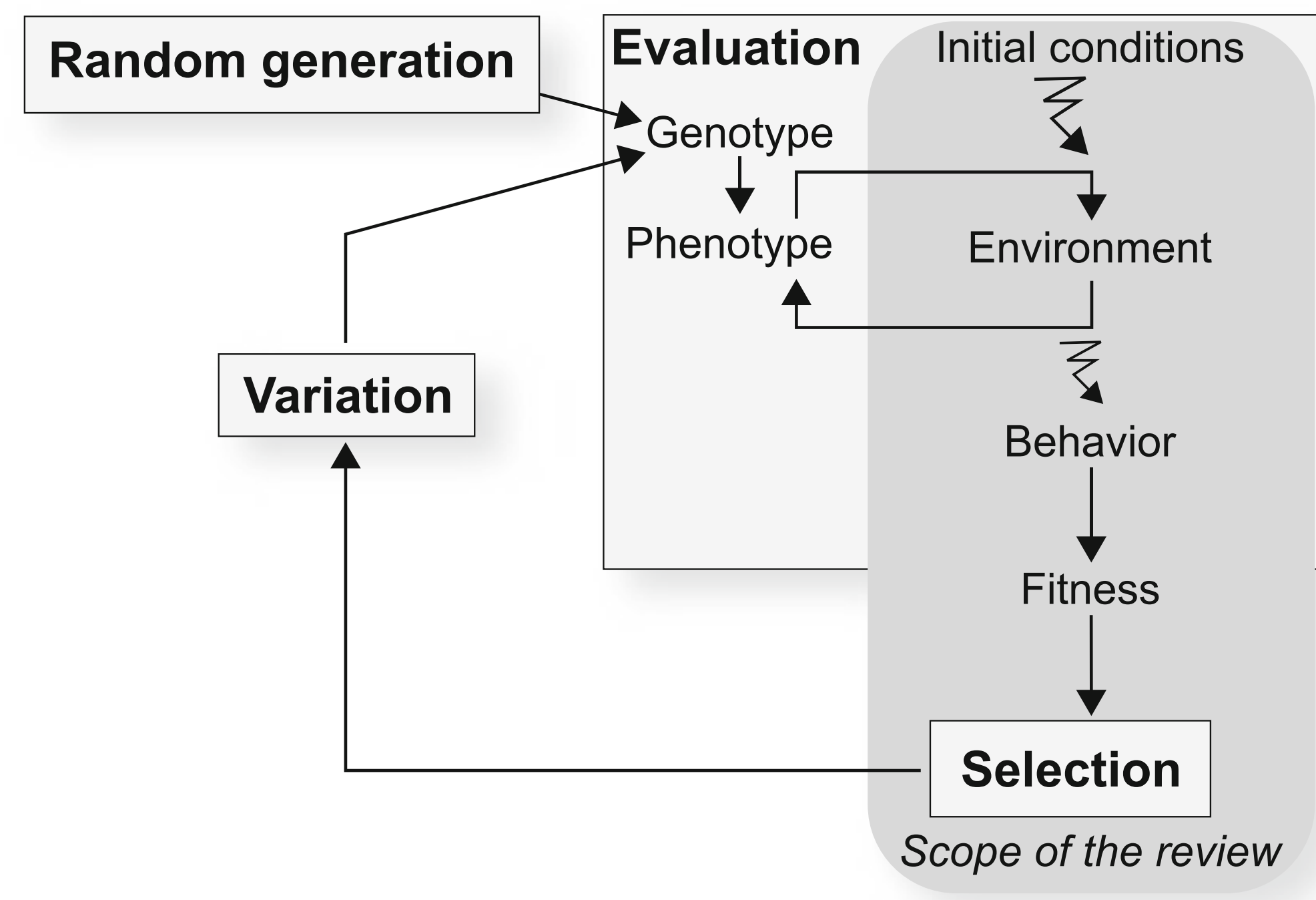
Stephane Doncieux · Jean-Baptiste Mouret

Selective Pressures in ER

A evolutionary black-box optimization

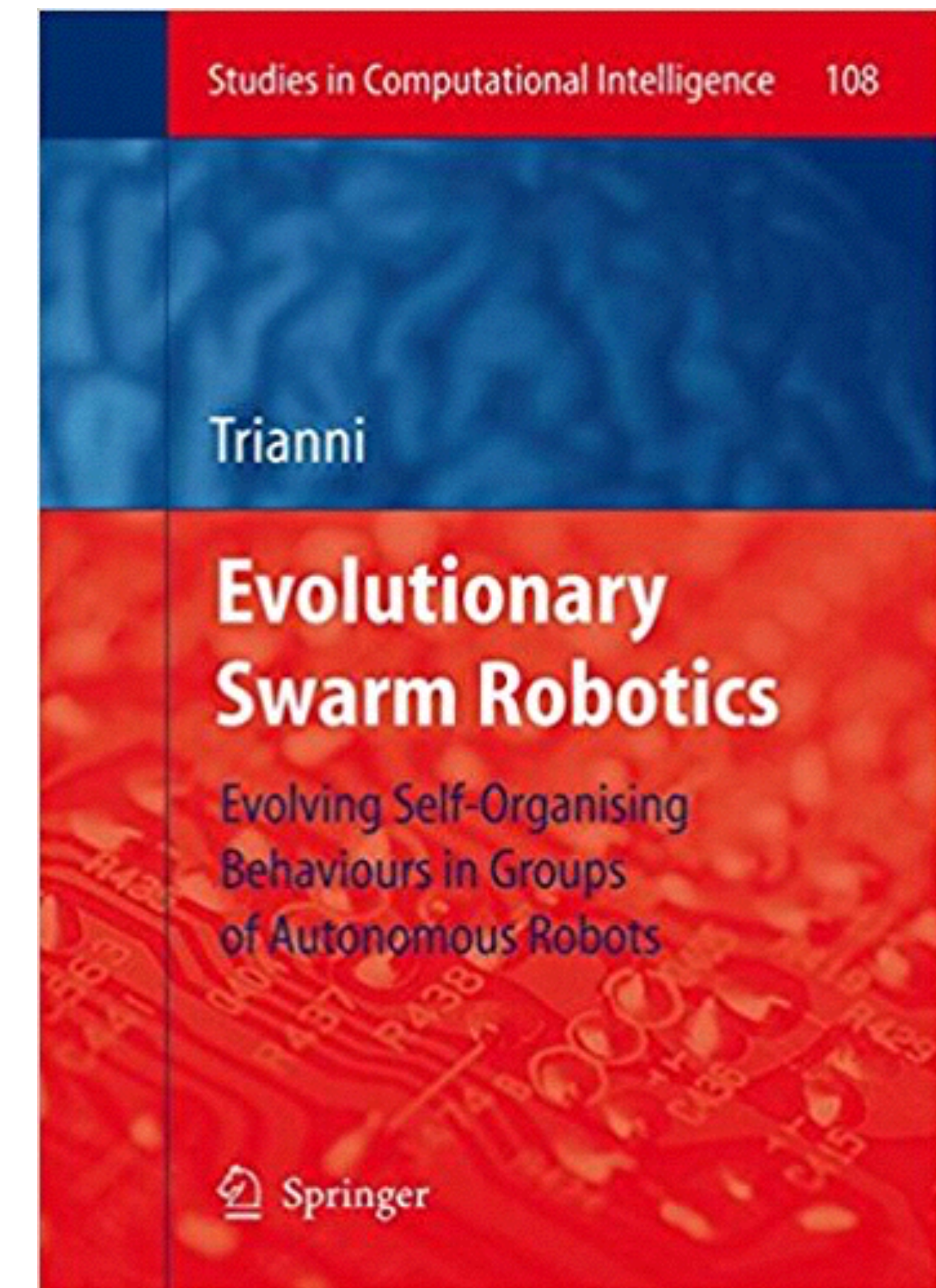


B evolutionary robotics



ER for Collective Behaviours

- Best fit for the design of collective behaviours (?)
- However, even **more design choices** are required!
- Indirect relationship between individual behaviours and group performance
- Existence of **conflicts** with heterogeneous groups



Required Design Choices

- Fitness function
- Genotype-to-phenotype mapping
- Sensory-motor configuration
- Stochastic sampling of the robot ecology

Artificial Life 17: 183–202 (2011)

**Engineering the Evolution of
Self-Organizing Behaviors
in Swarm Robotics:
A Case Study**

Vito Trianni^{*,**}

Université Libre de Bruxelles

Stefano Nolfi[†]

Istituto di Scienze e Tecnologie
della Cognizione

Rule #1: Use multiple objectives

Rule #1: Use multiple objectives

- Defining a good fitness function is not easy
 - Encoding desired features of a behaviour in a measurable form
 - introduction of multiple **behavioural terms**
 - Ensuring evolvability of the system
 - avoid to remain trapped in **local minima**
 - avoid **bootstrap** problems
- **Multi-Objective Optimisation (MOO)** provides tools to deal with this
- How does MOO compares with Single-Objective Optimisation (SOO)?

Case Study: Flocking

- A genuinely multi-objective problem

- **Cohesion:** stay together

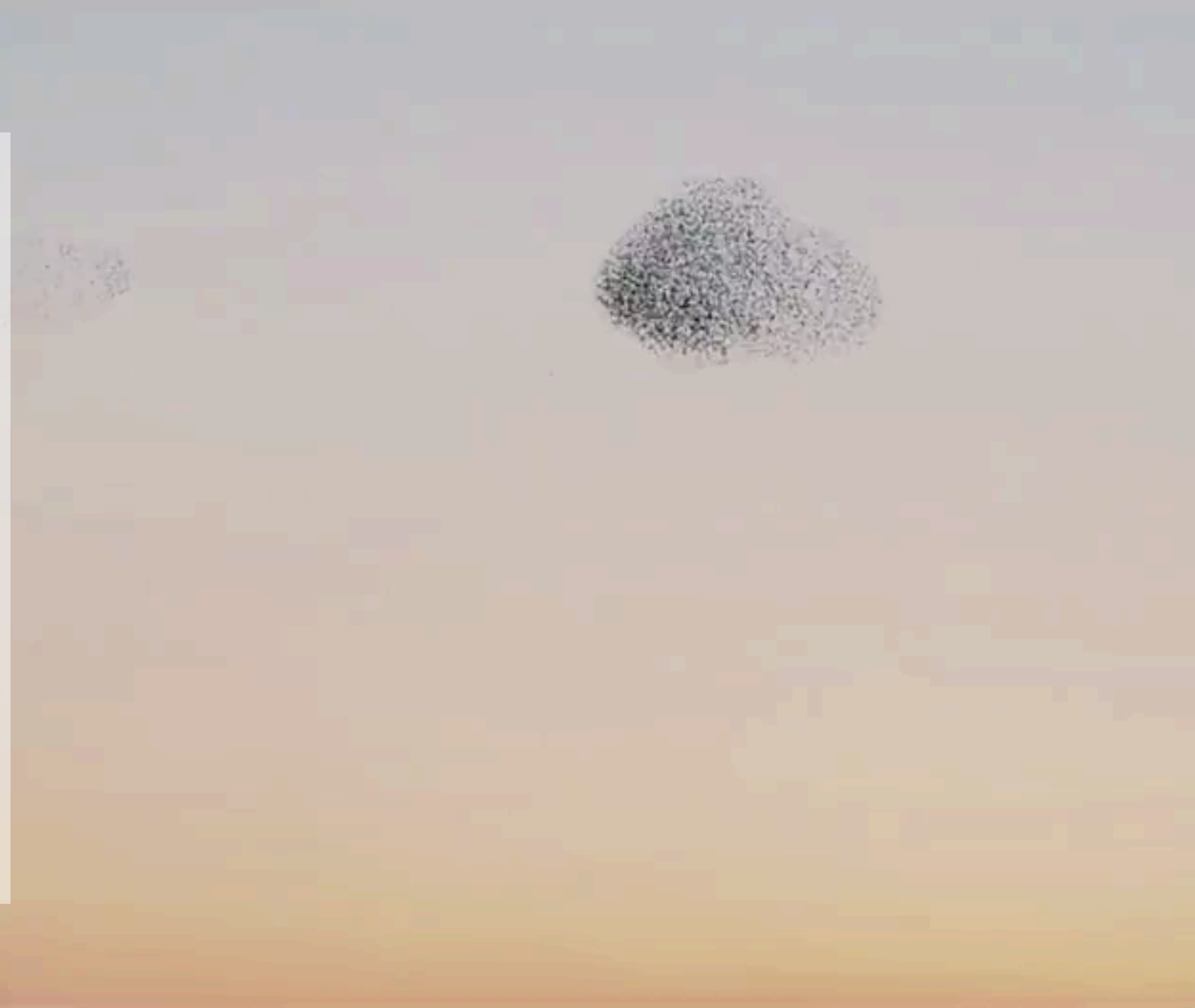
$$\mathcal{F}_m = \frac{\|\hat{\mathbf{X}}(T) - \hat{\mathbf{X}}(0)\|}{D_m(T)}$$

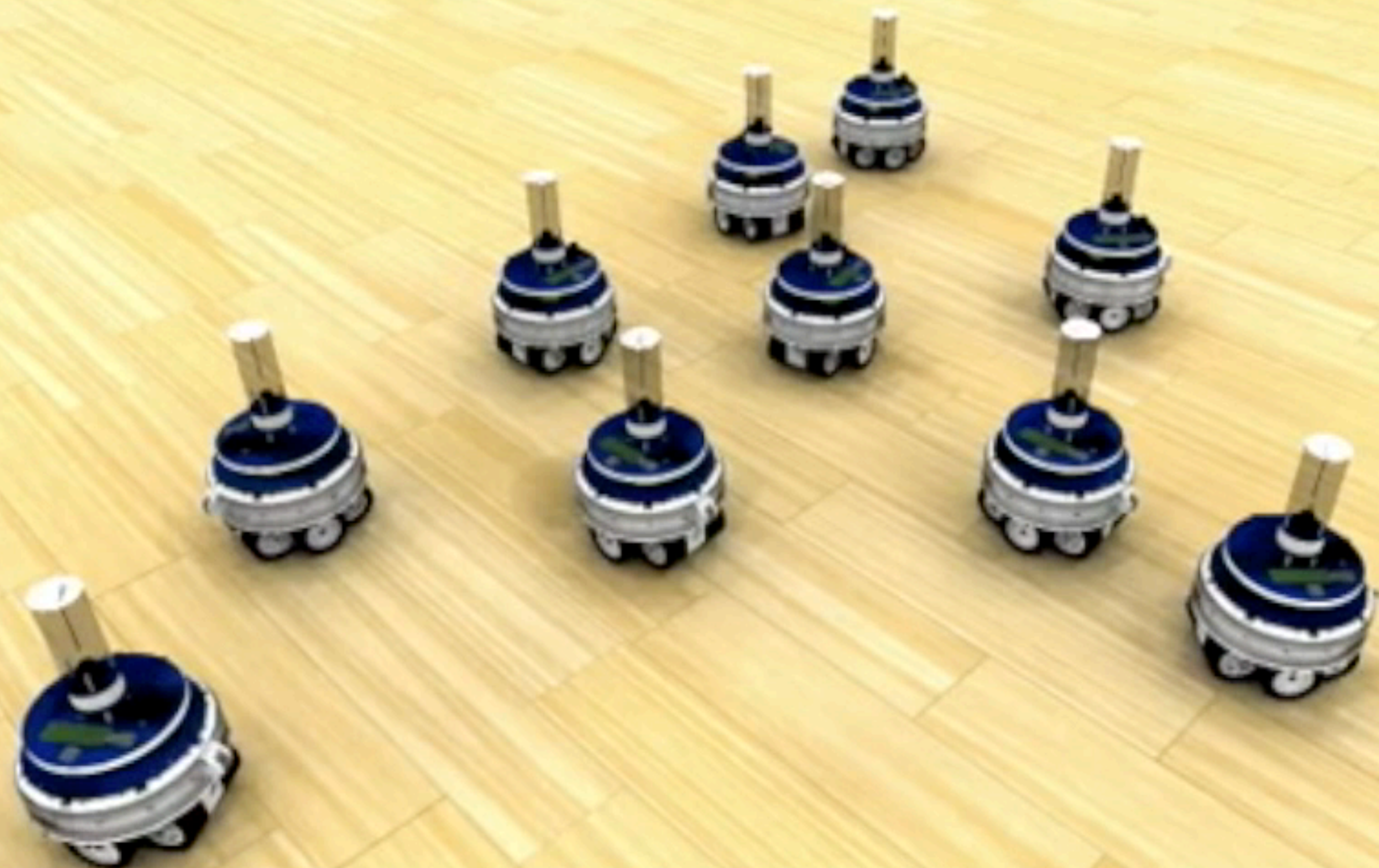
- **Motion:** move as much as possible

$$\mathcal{F}_c = \max \left(0, 1 - \frac{1}{N} \sum_i \frac{\|\mathbf{X}_i(T) - \hat{\mathbf{X}}(T)\|}{d_m} \right)$$

- Single objective from scalarization

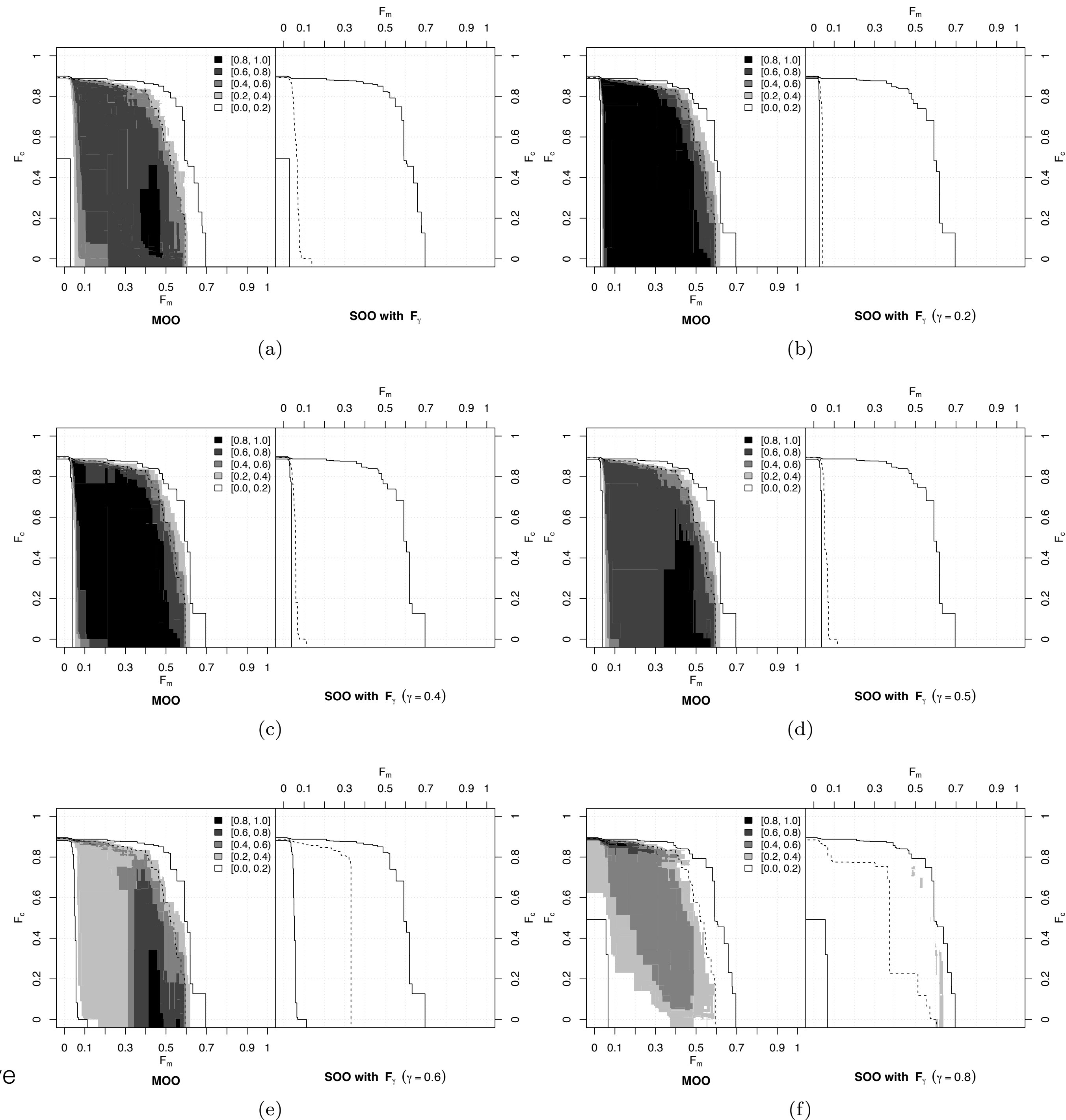
$$\mathcal{F}_\gamma = \gamma \mathcal{F}_m + (1 - \gamma) \mathcal{F}_c$$





Results

- We compare MOO with SOO through empirical attainment functions (EAFs)
- MOO performs better than SOO
- MOO produces good flocking in every single run
- MOO produces varied behaviours



Case study: a strictly collaborative task

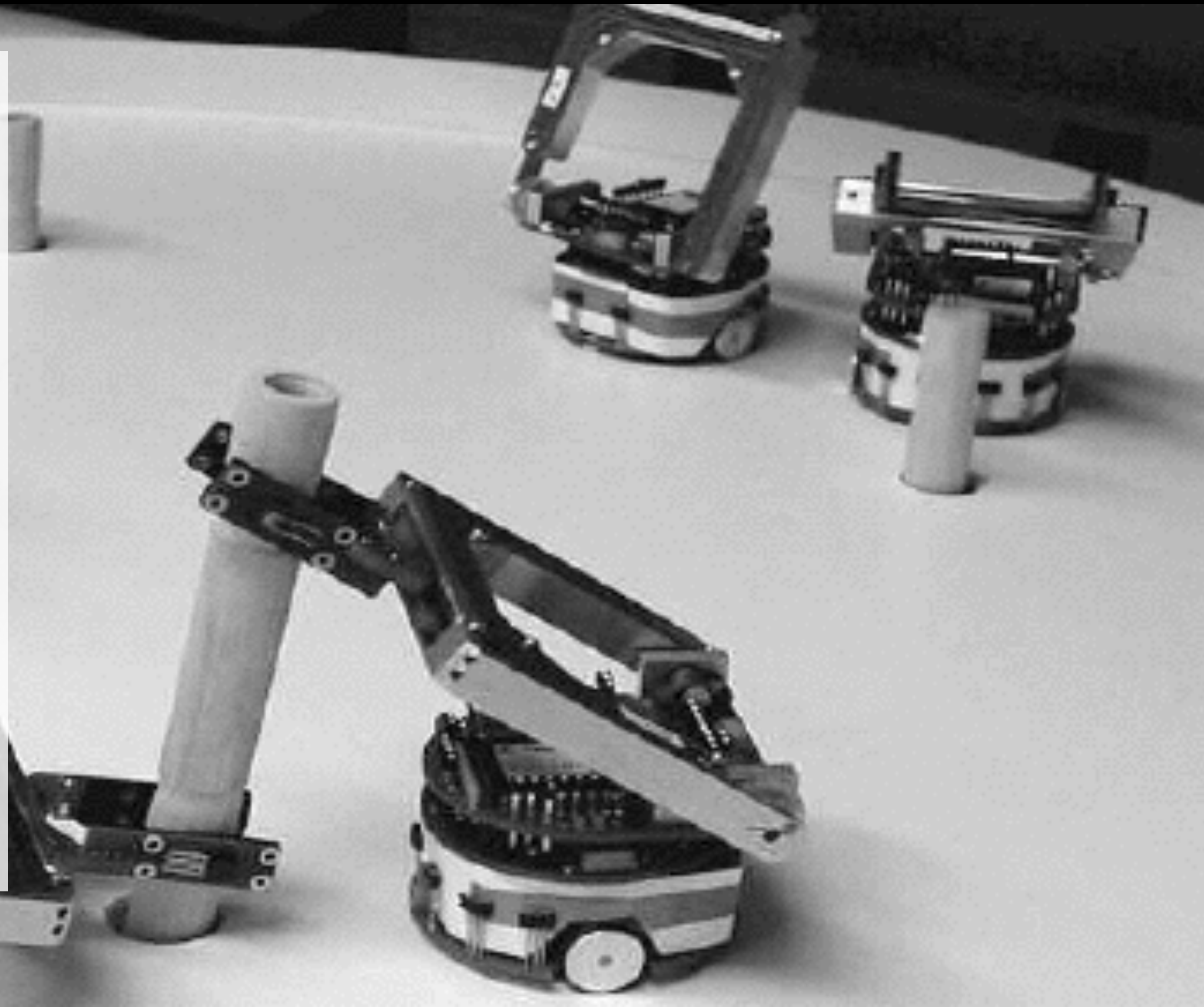
- Collaboration is needed to accomplish the task
- Complexity given by the number of robots required per collaboration

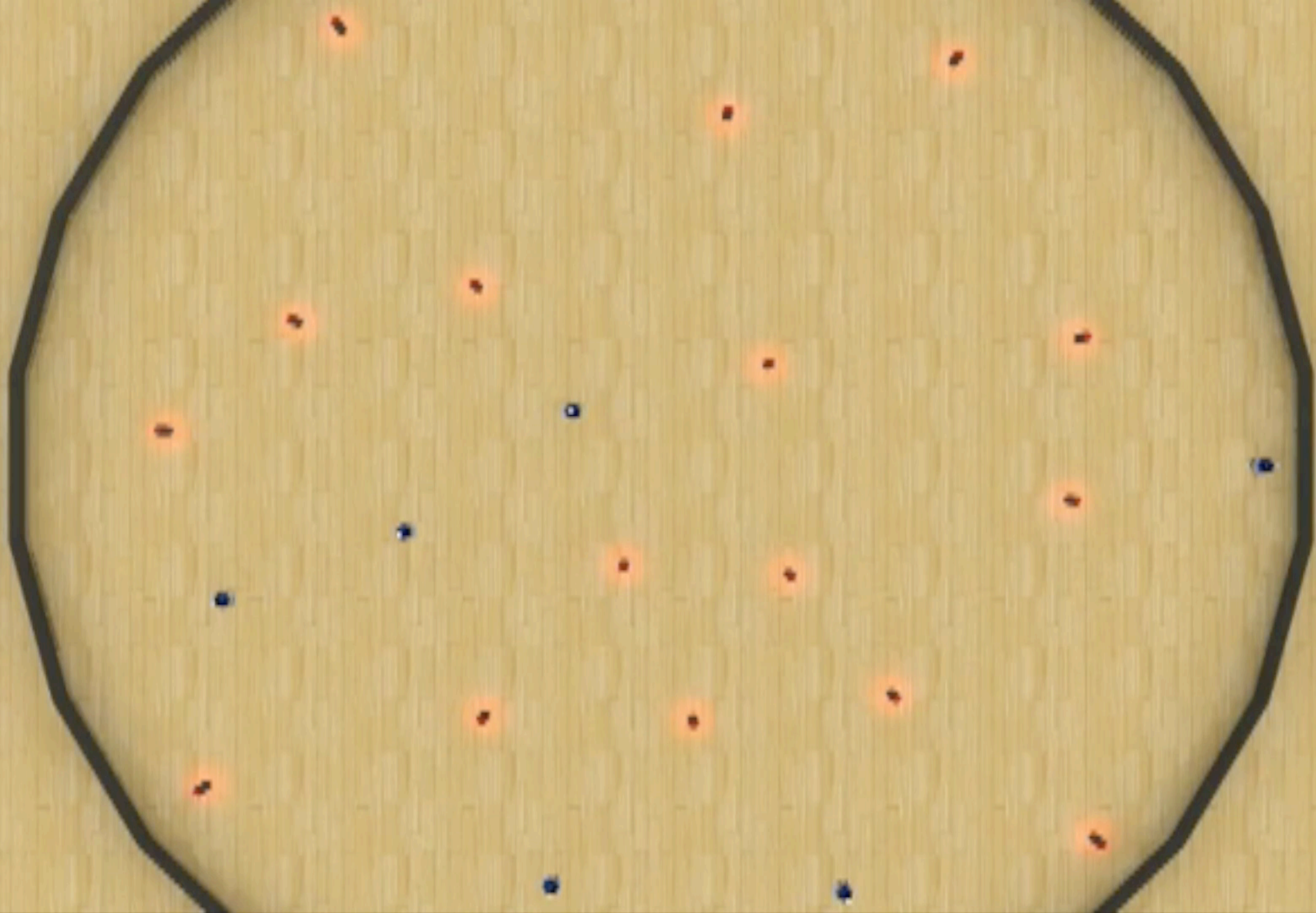
- SOO: number of collaborations:

$$\mathcal{S}_c = \frac{|C(T)|}{C_M}$$

- Ancillary objective: average number of visited collaboration sites”

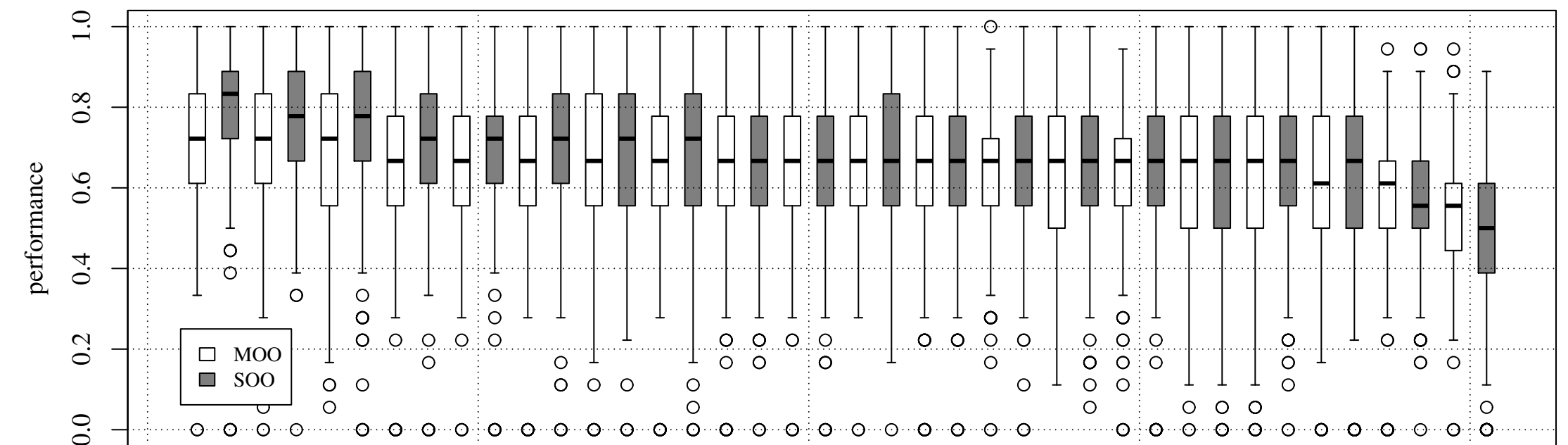
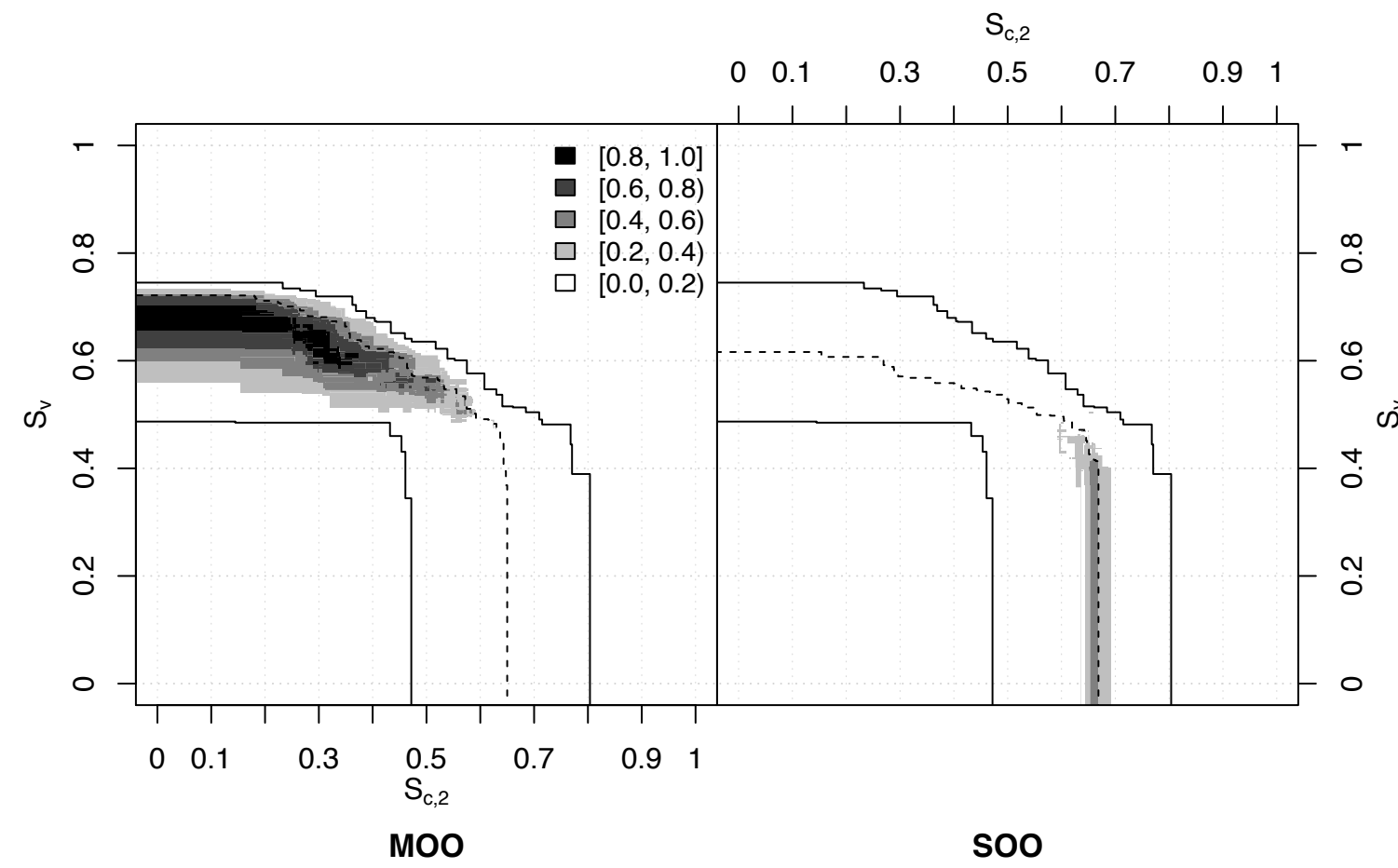
$$\mathcal{S}_v = \frac{1}{N} \sum_i \frac{|V_i(T)|}{C_M}$$



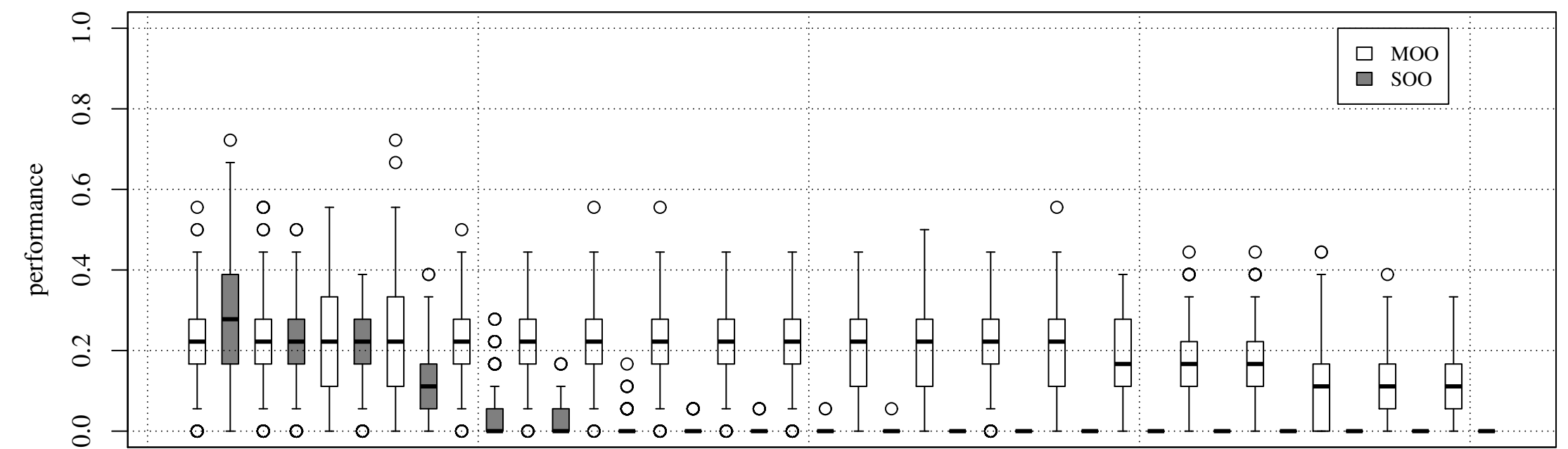
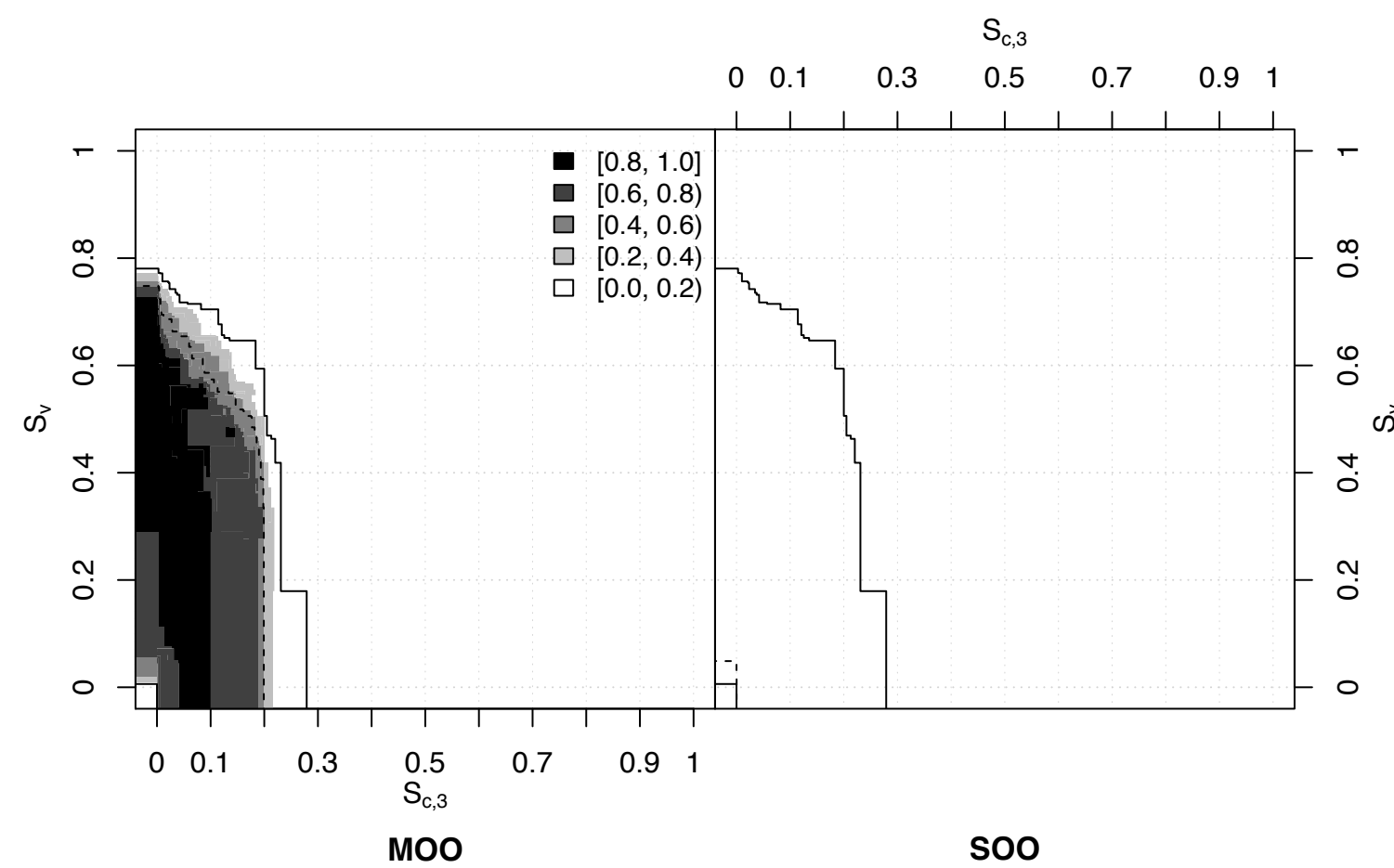


Results

CL: 2 robots →



CL: 3 robots →



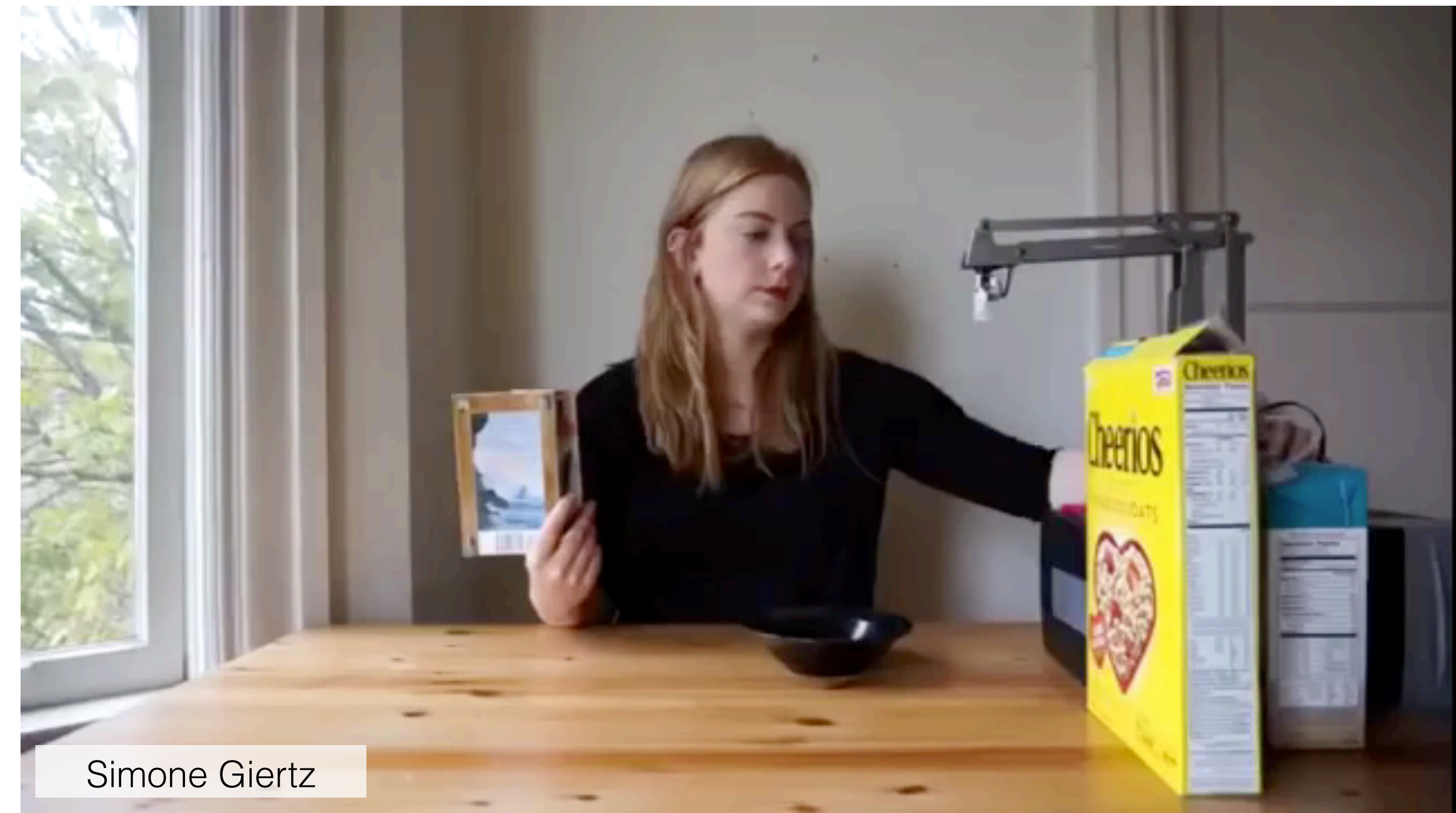
Take-Home Message #1

- In ER, the **behaviour** has primacy
- Fitness serves to drive the search for a satisfactory solution
- SOO perform best when an **evolvable fitness** is available
- MOO provides
 - large exploration of the space of possible solutions
 - better evolvability

Rule #2: Carefully configure your robots

Rule #2: Carefully configure your robots

- Even for well-defined task, choosing the right robot configuration can make the difference
- In collective robotics, **interactions** are fundamental
- How to design the right way to interact among neighbours?



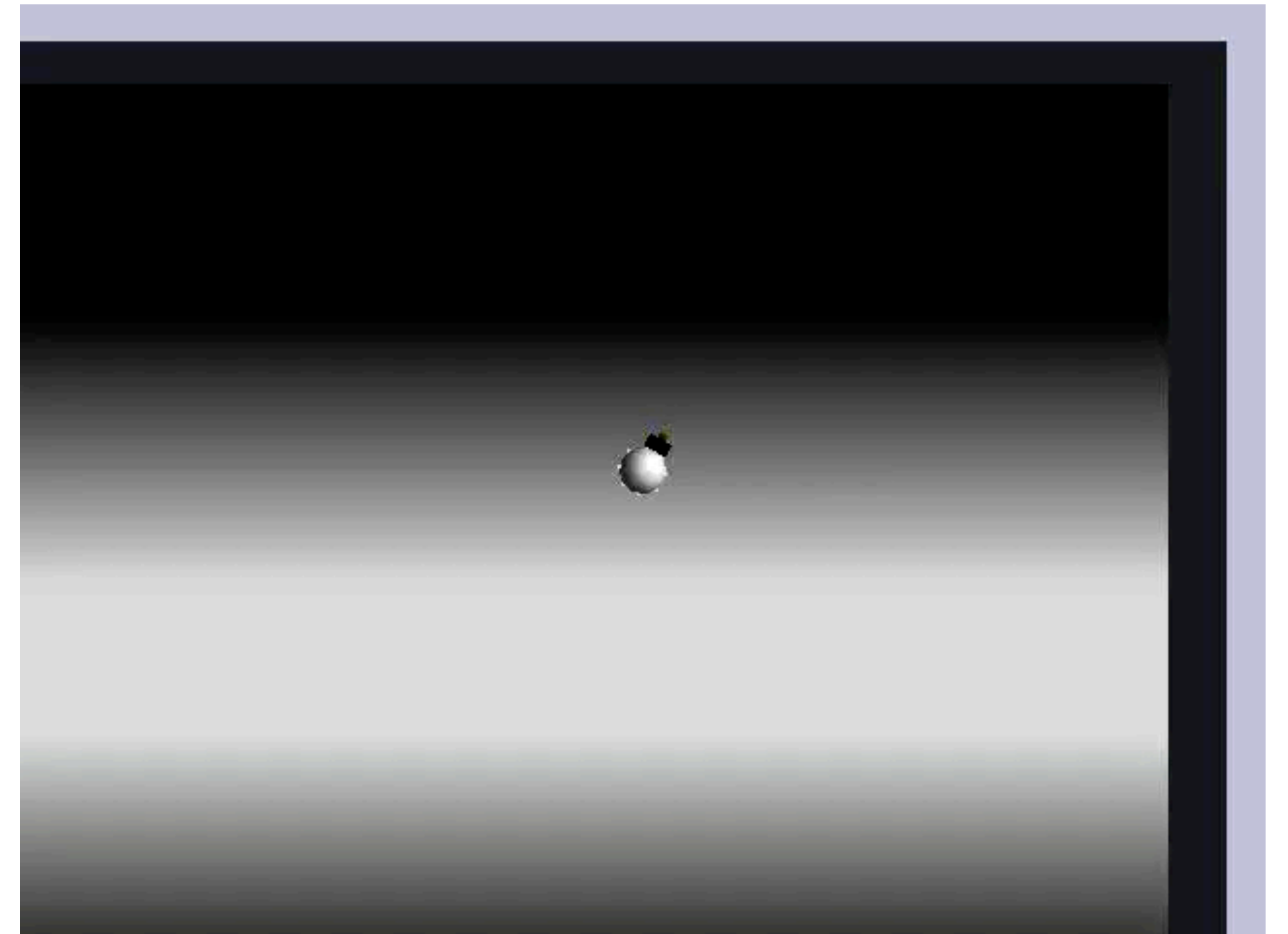
Case Study: Synchronisation

- Basic behaviour for swarm robotics



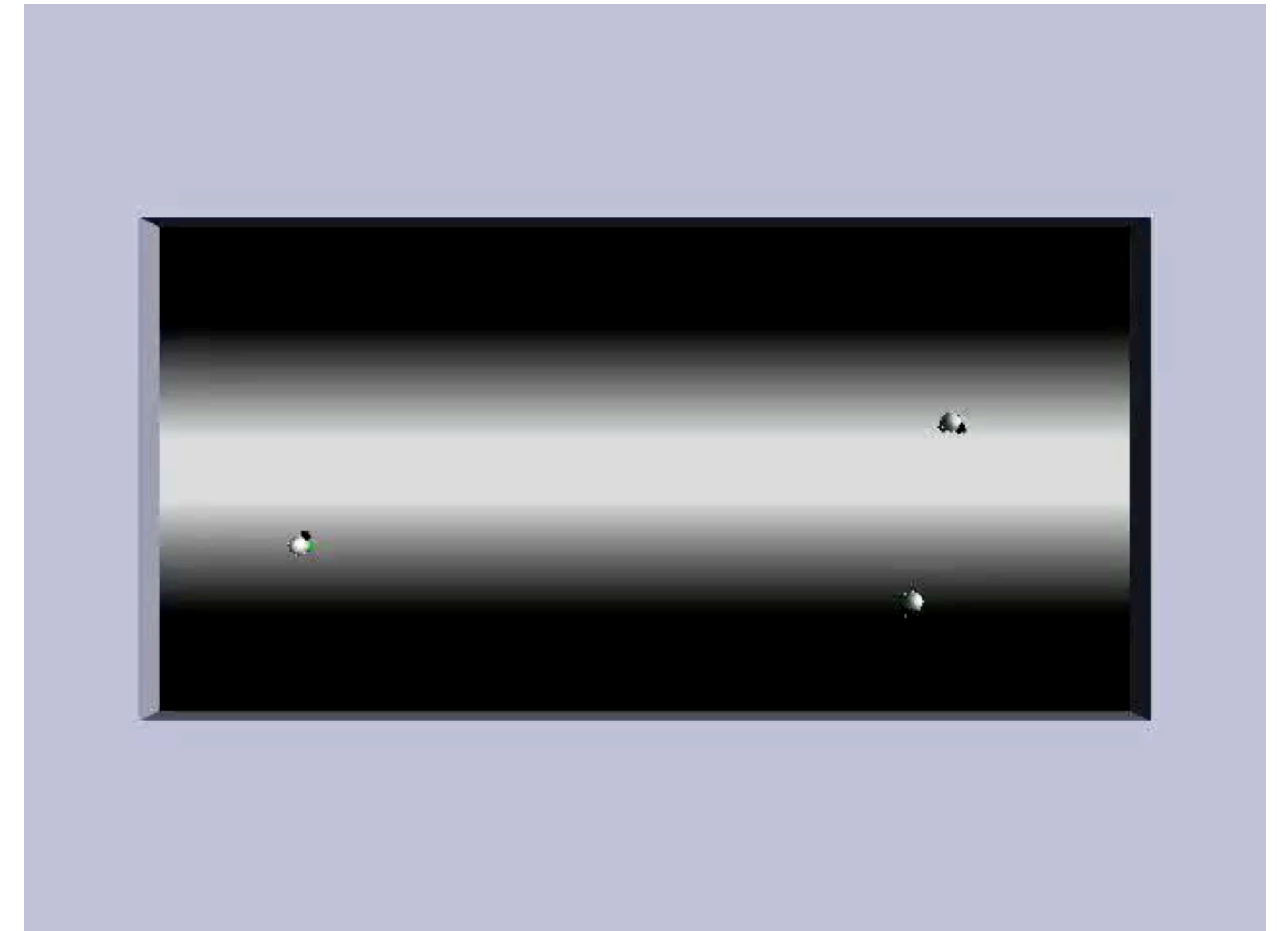
Case Study: Synchronisation

- Basic behaviour for swarm robotics
- Sync of periodic robot behaviours



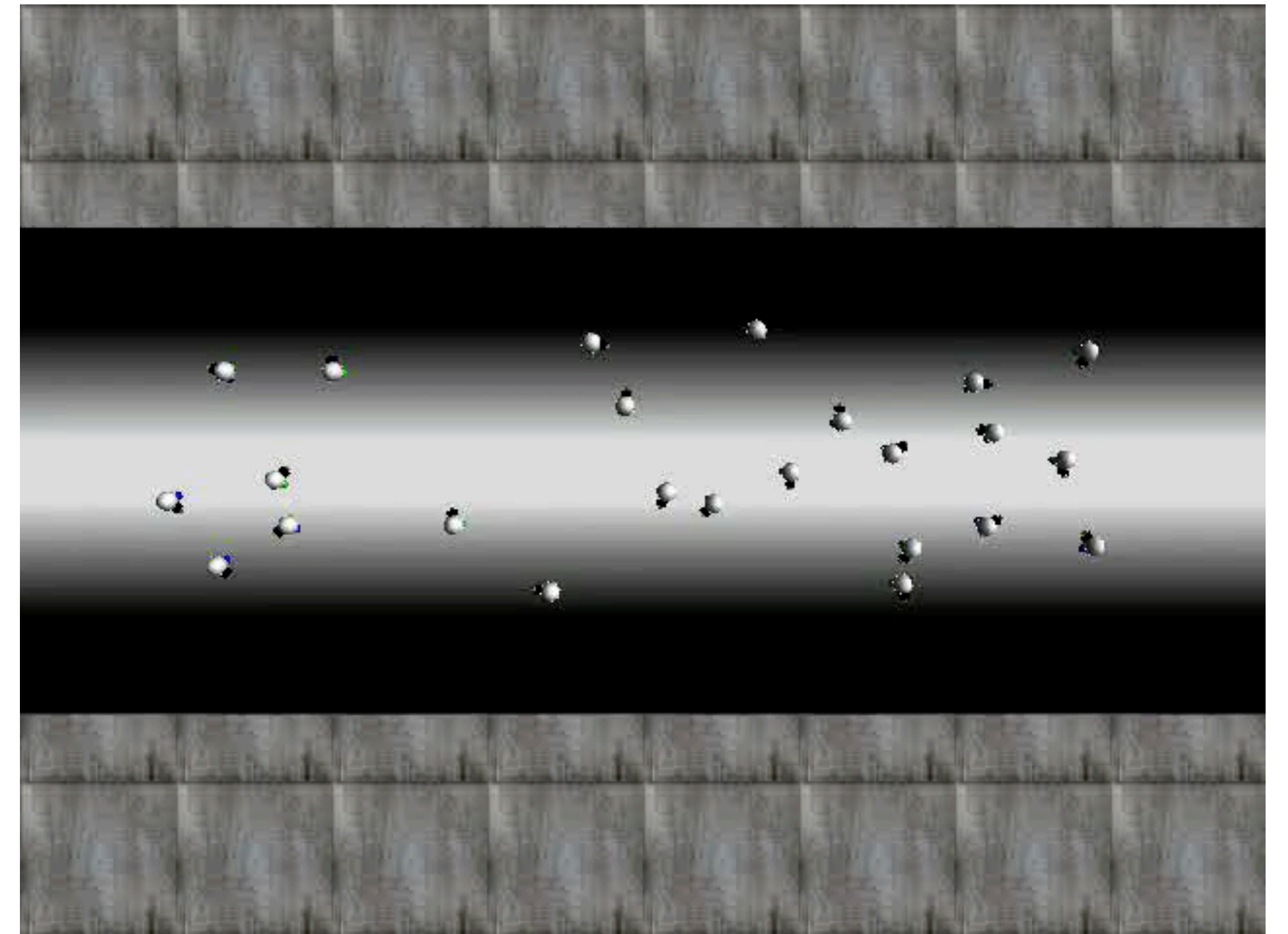
Case Study: Synchronisation

- Basic behaviour for swarm robotics
- Sync of periodic robot behaviours
- **Signalling** is required for sync



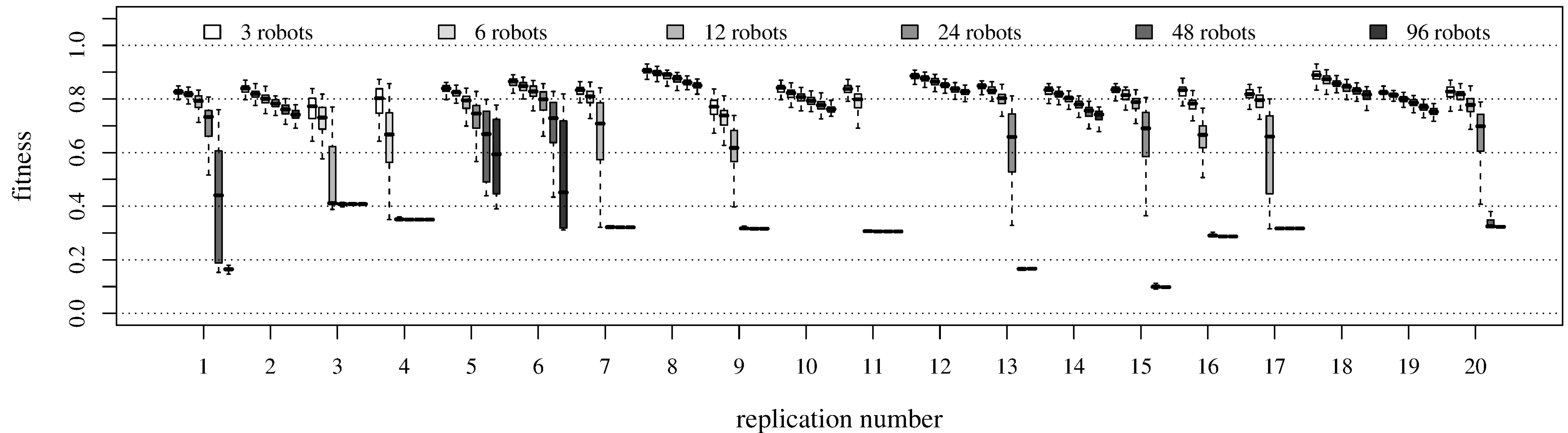
Case Study: Synchronisation

- Basic behaviour for swarm robotics
- Sync of periodic robot behaviours
- **Signalling** is required for sync
- Poor design of communication prevents scalability



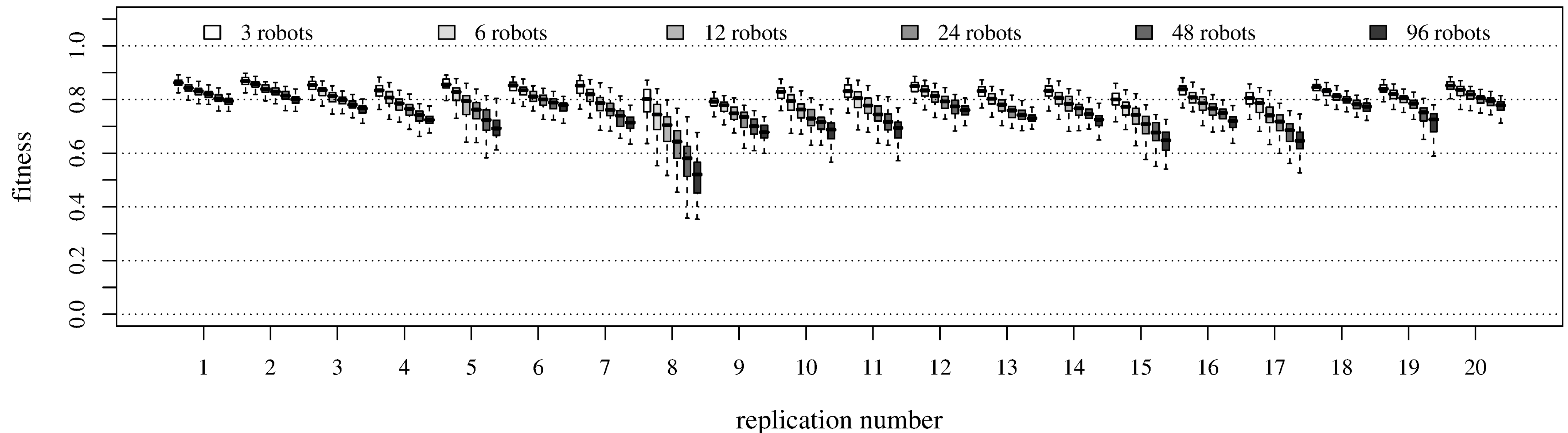
Results

- Binary signals leads to interference with large groups



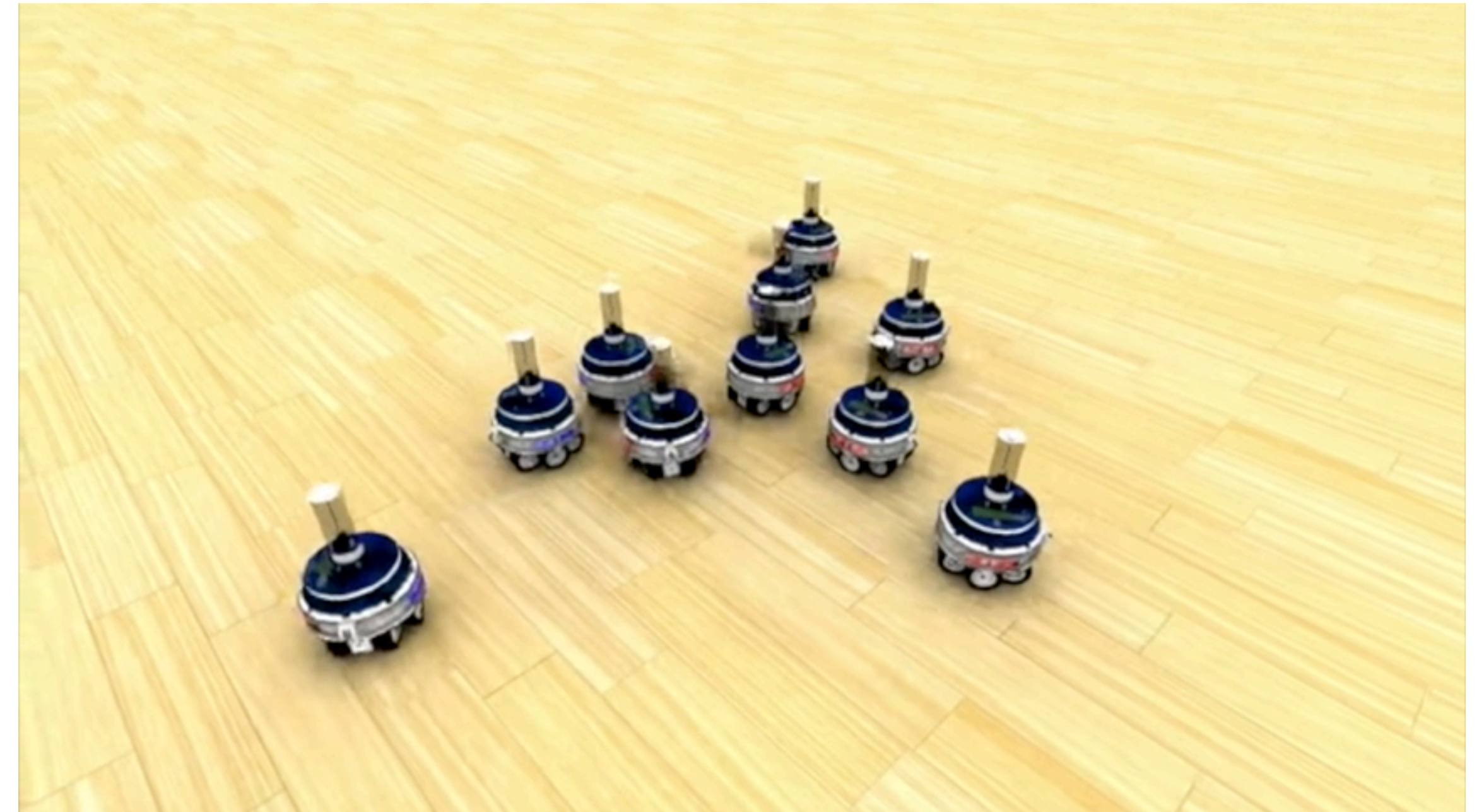
Results

- Binary signals leads to interference with large groups
- Introduction of intensity-based signal perception leads to scalability



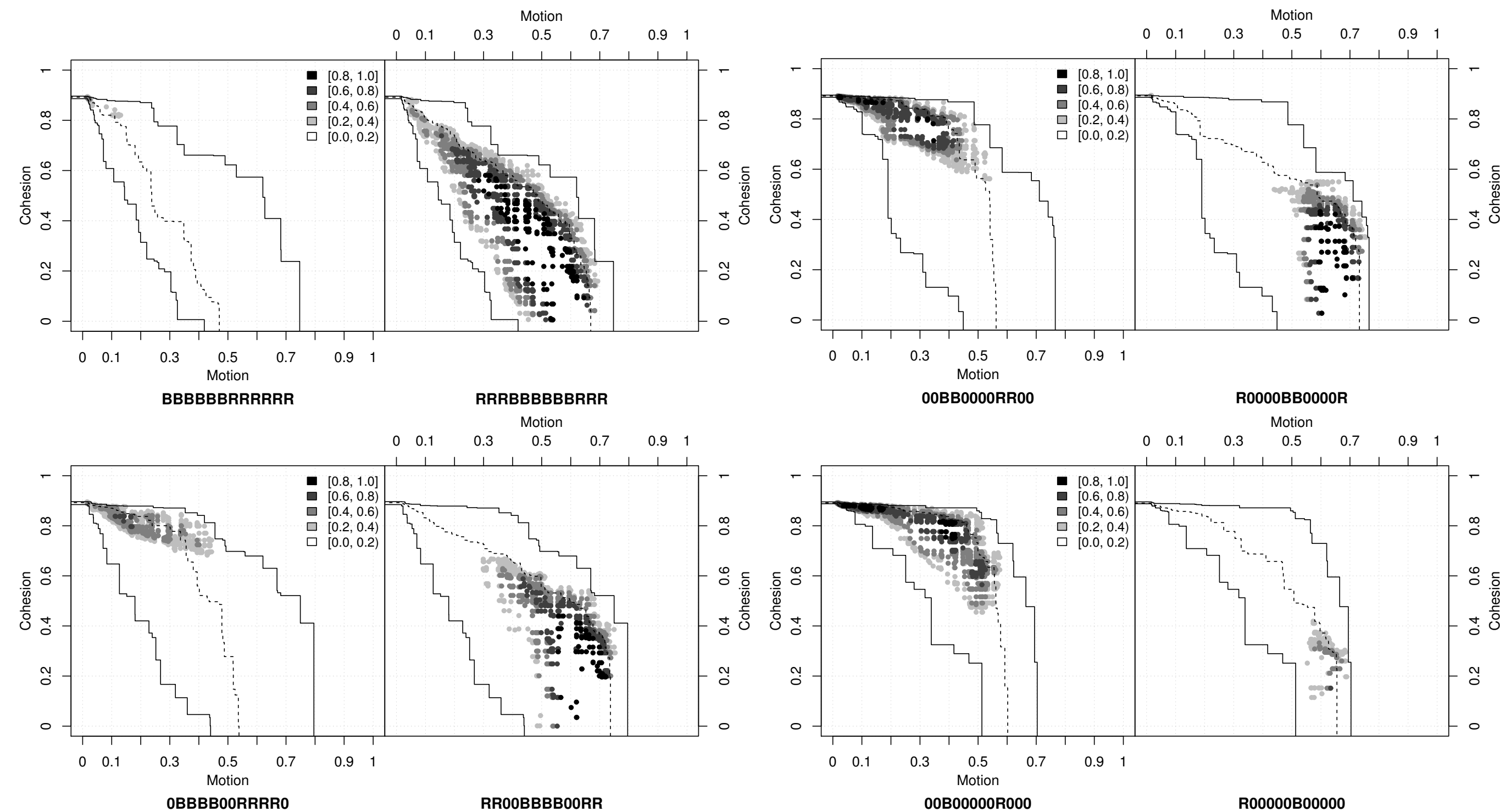
Case Study: Flocking

- Use coloured LEDs to display the robot orientation
- Tested many LEDs patterns
 - Left-right
 - Front-rear
 - Uniform (for control)



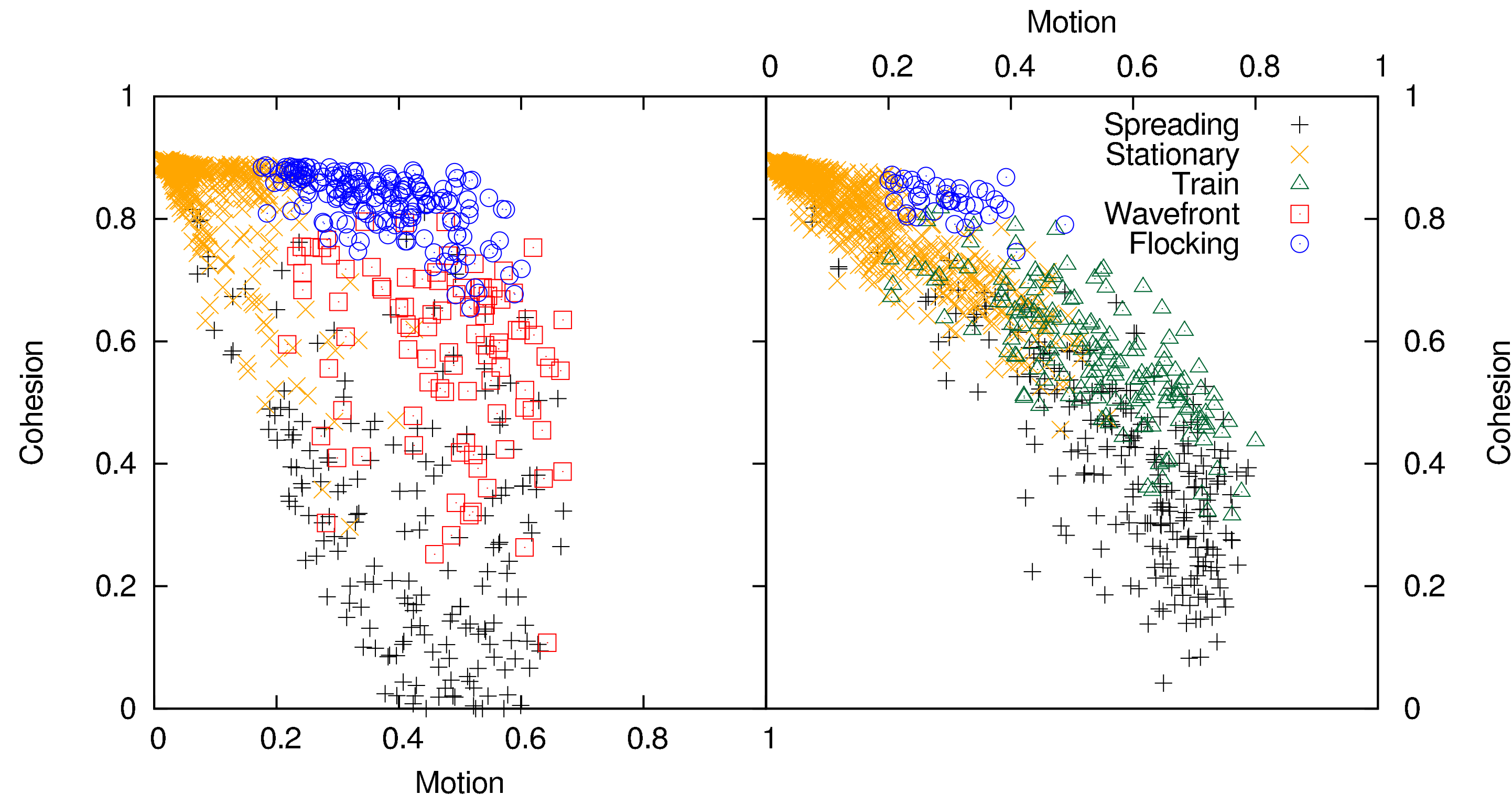
Case Study: Flocking

- Use coloured LEDs to display the robot orientation
- Tested many LEDs patterns
 - Left-right
 - Front-rear
 - Uniform (for control)
- Efficiency strongly affected



Case Study: Flocking

- Use coloured LEDs to display the robot orientation
- Tested many LEDs patterns
 - Left-right
 - Front-rear
 - Uniform (for control)
- Efficiency strongly affected
- Resulting behaviour depends on the LEDs pattern



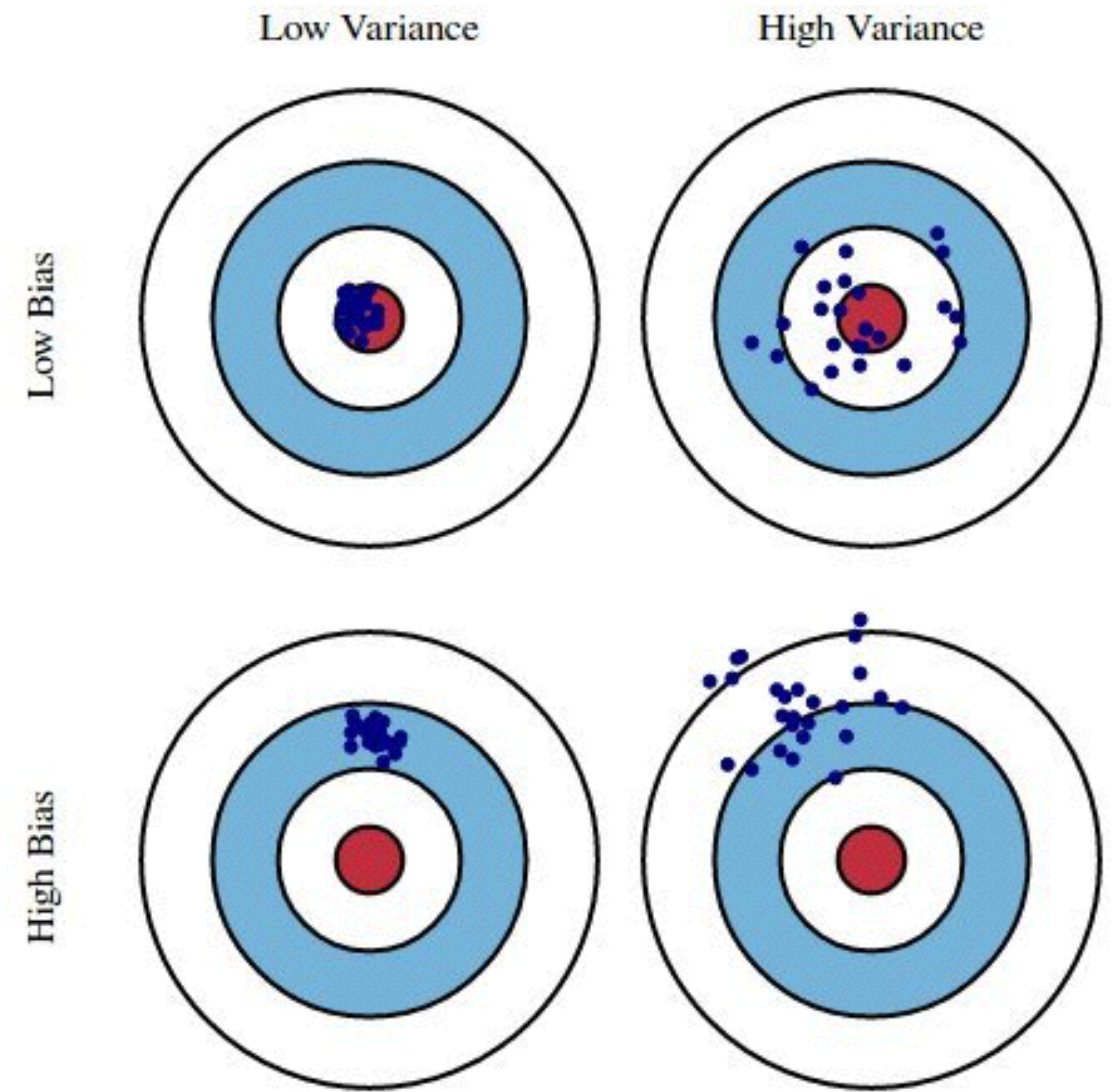
Take-Home Message #2

- In collective behaviours, **interactions** are fundamental
- Intuitive choices may be strongly misleading
- ER can deal with oddly designed interaction modalities, but **evolvability** or **scalability** may be affected
- Test **different modalities** for their effects, or put the robot **configuration under evolution** (with MOO?)

Rule #3: Mind the bias-variance tradeoff

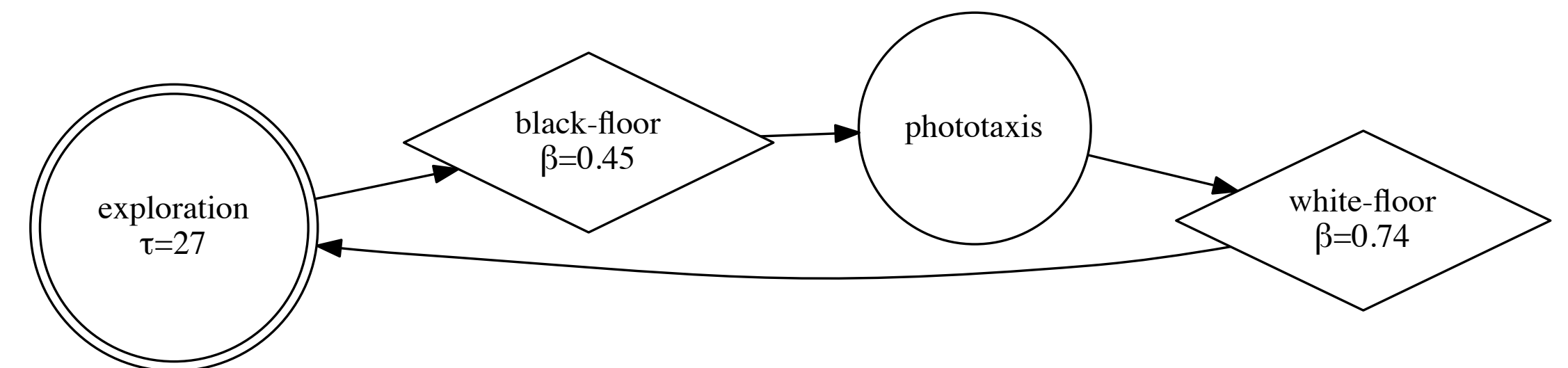
Rule #3: Mind the bias-variance tradeoff

- Powerful representations like NNs are characterised by
 - **low bias**: potentially approximate any function
 - **high variance**: overfit the training set, difficulties in generalisation (*reality gap*)
- Available techniques do not scale well to collective robotics
- Solution: **inject bias** in the control architecture
- Approach: use **modular control architectures** and indirect encodings



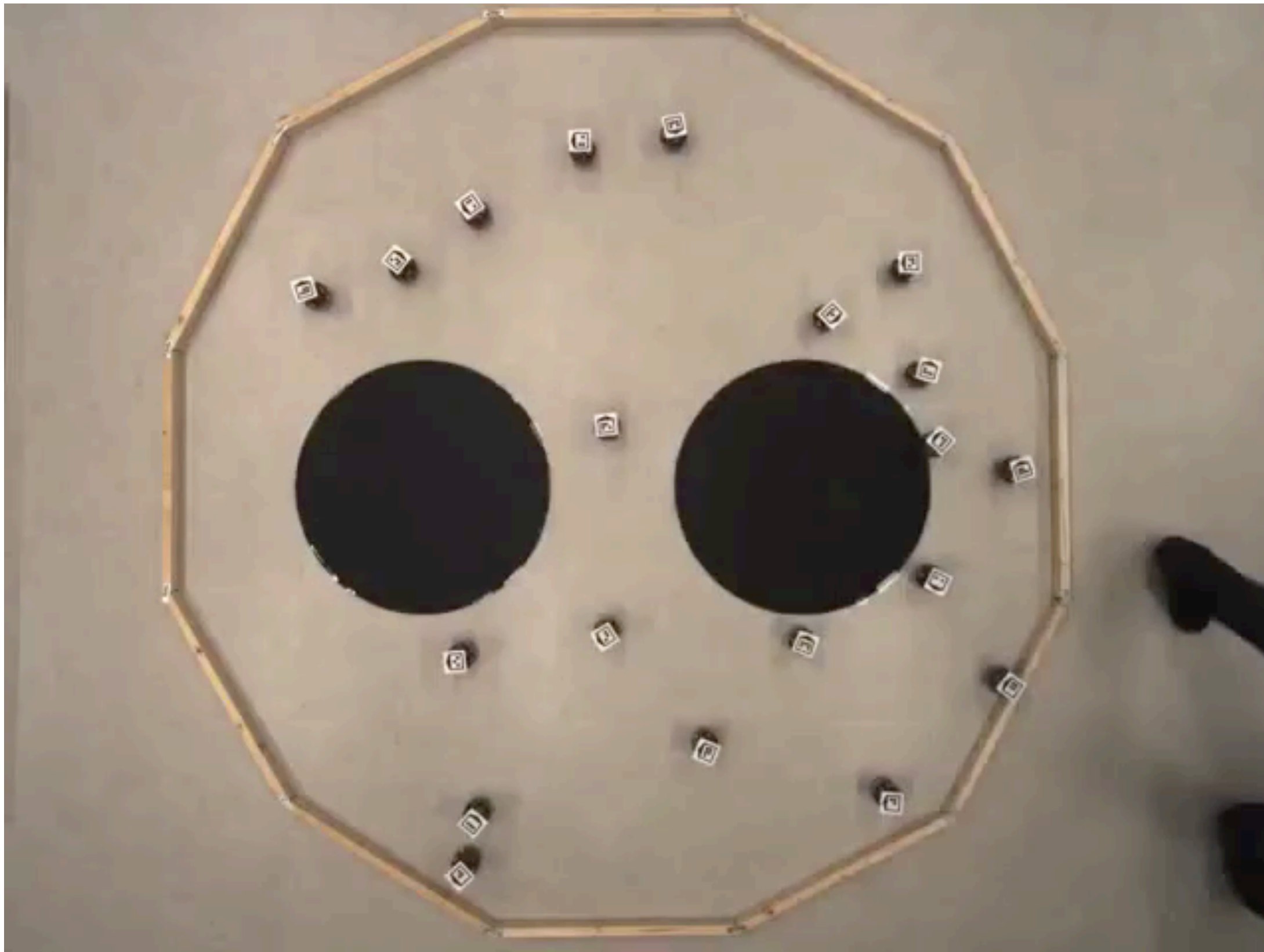
Automatic Modular Design

- **AutoMoDe**: Synthesis of probabilistic finite state machines (PFSMs)
 - States represent **constituent behaviours** with parameters for fine-tuning
 - Transitions represent **conditions** upon which to change the current behaviour
- A **reference model** is introduced to describe the robot capabilities
- **Optimisation** is performed on the space of possible PFSMs

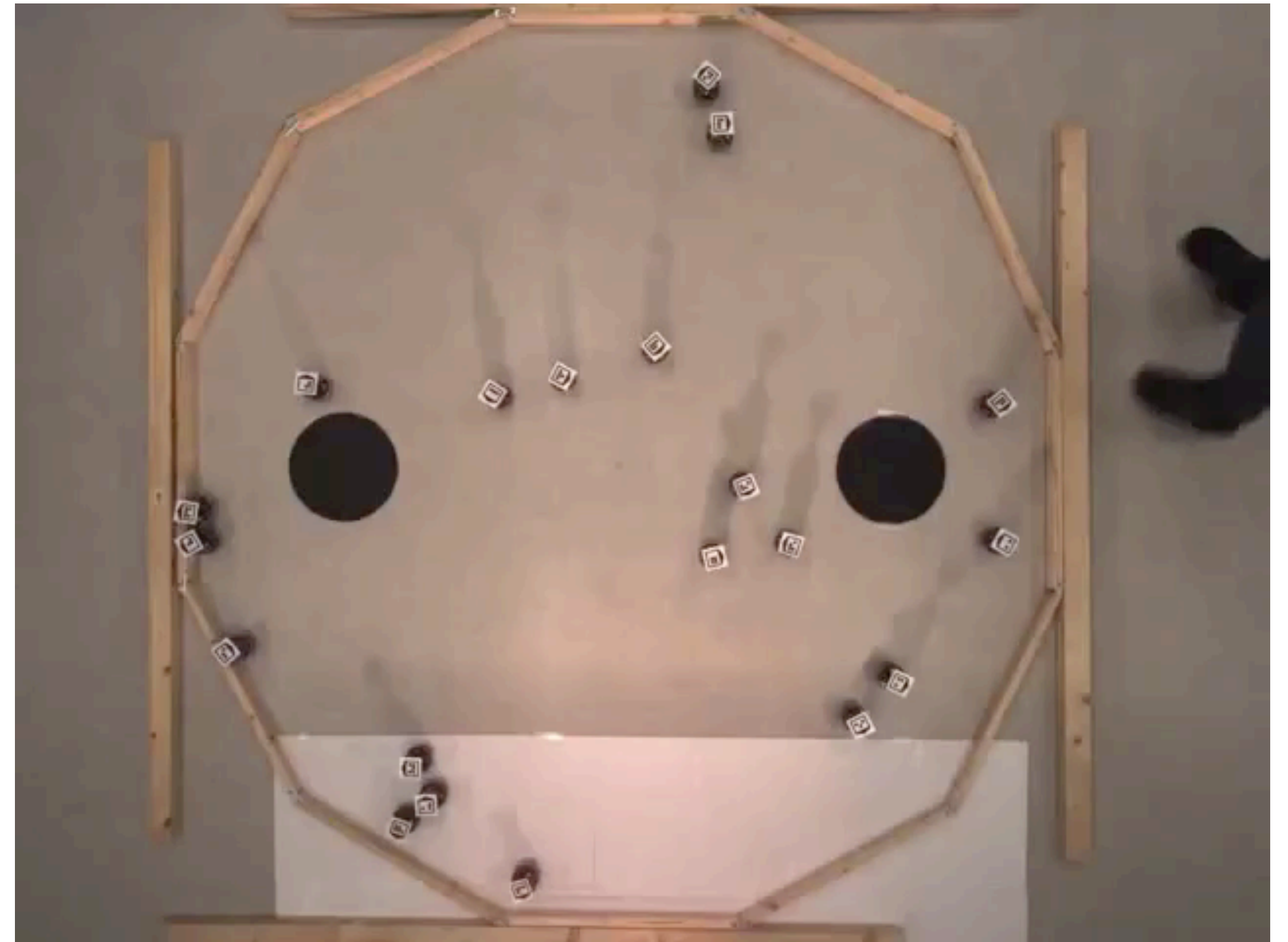


Results

Aggregation



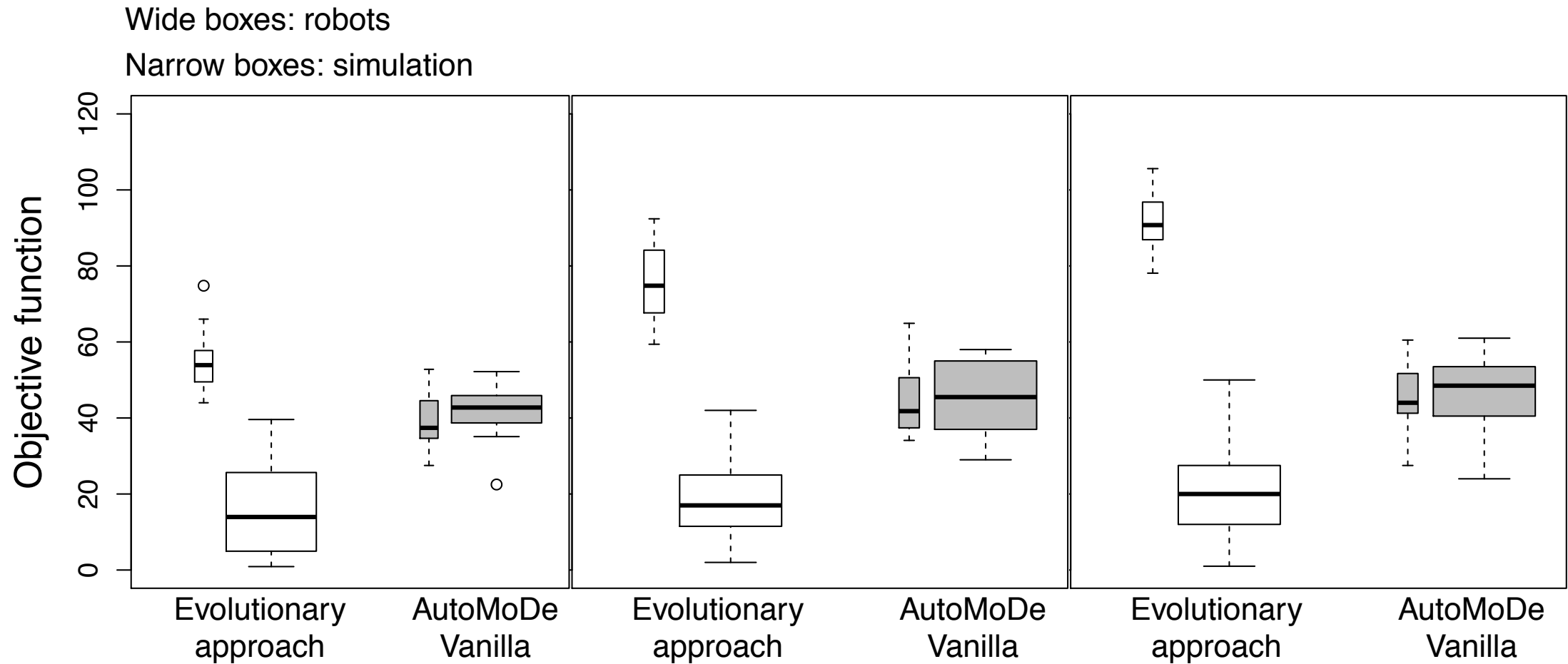
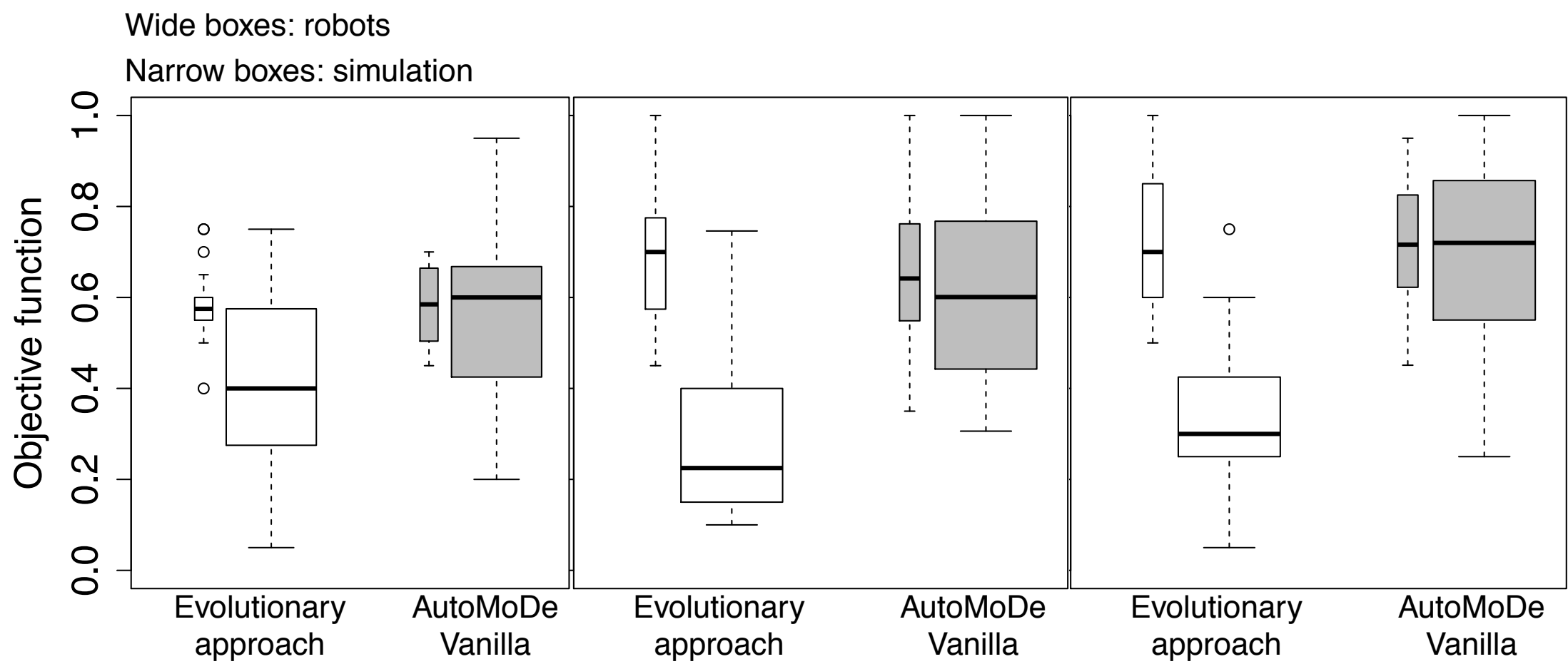
Foraging



Results

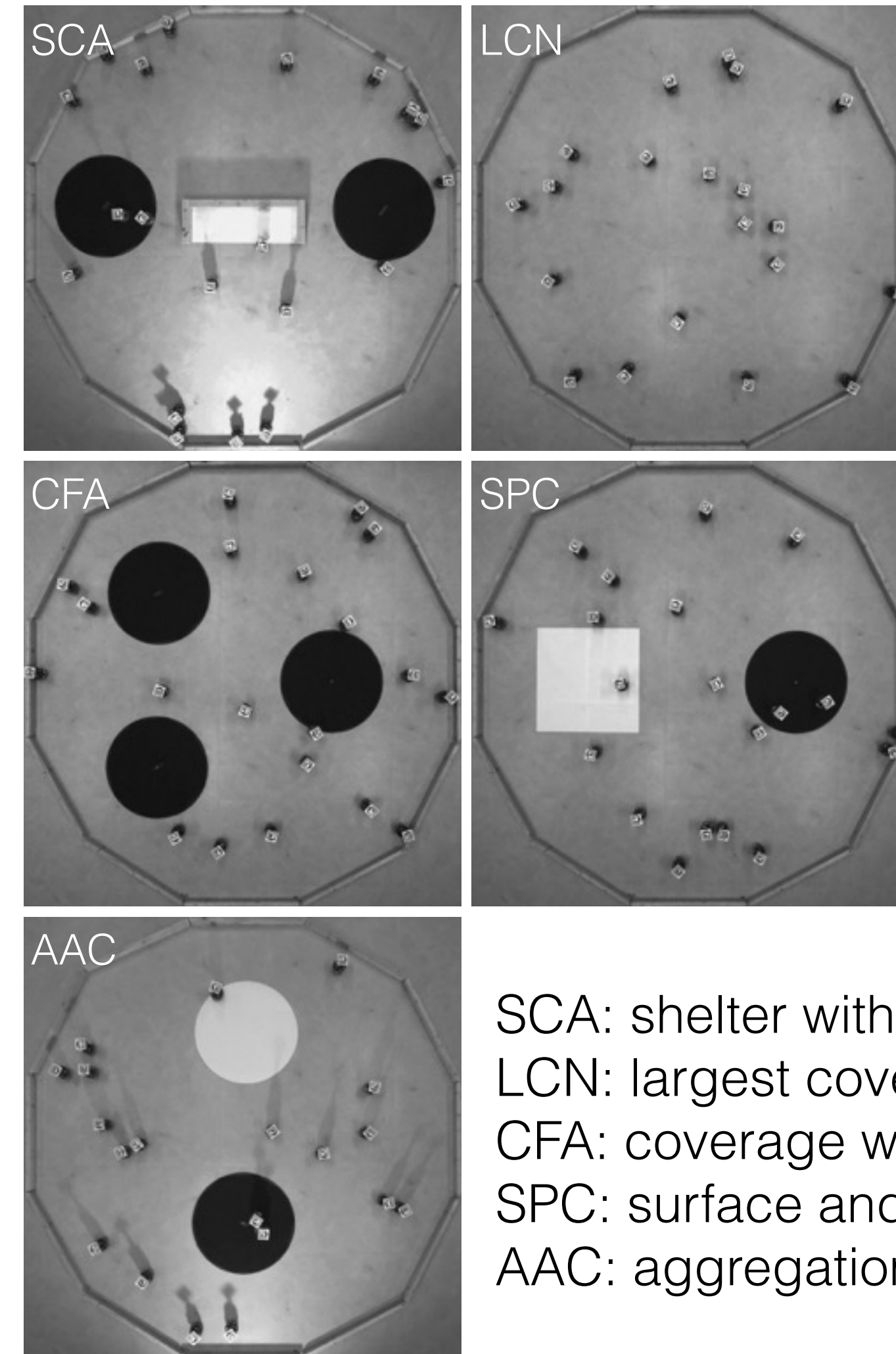
Aggregation

Foraging



Comparing with Human Designs

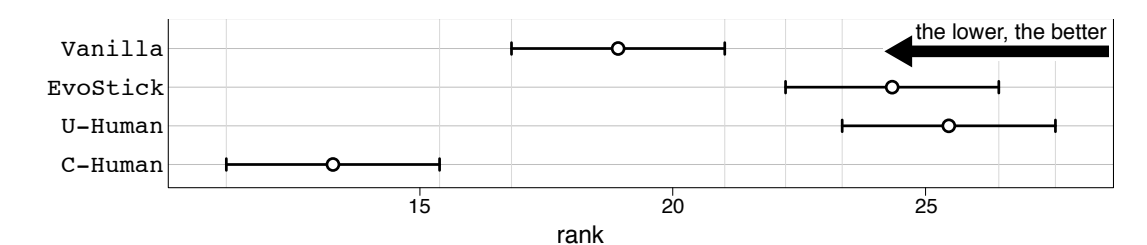
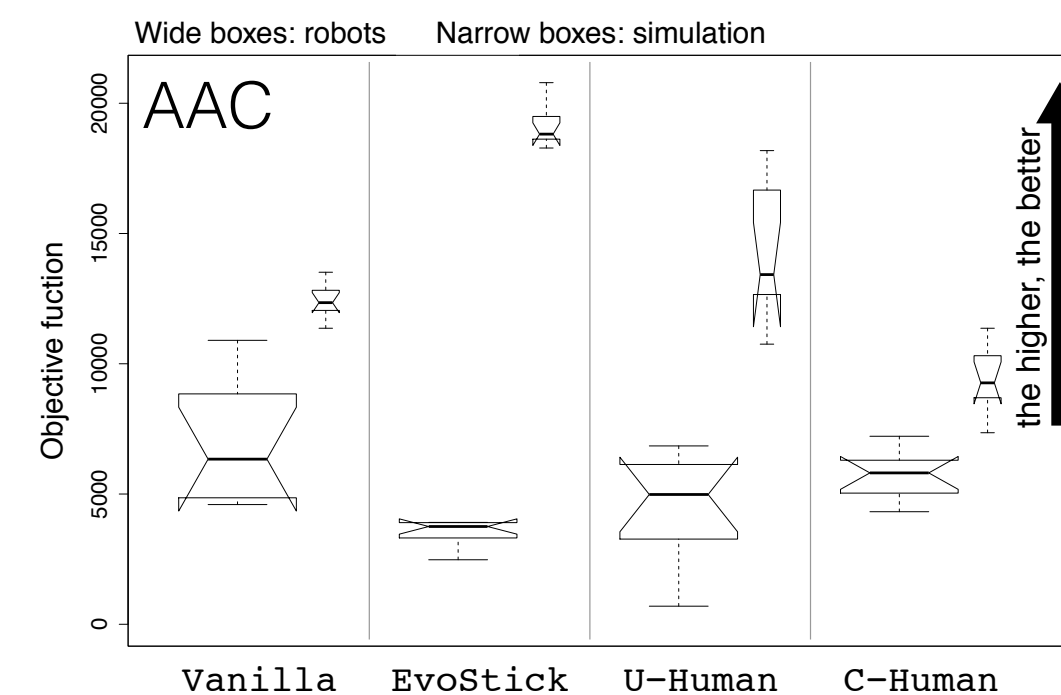
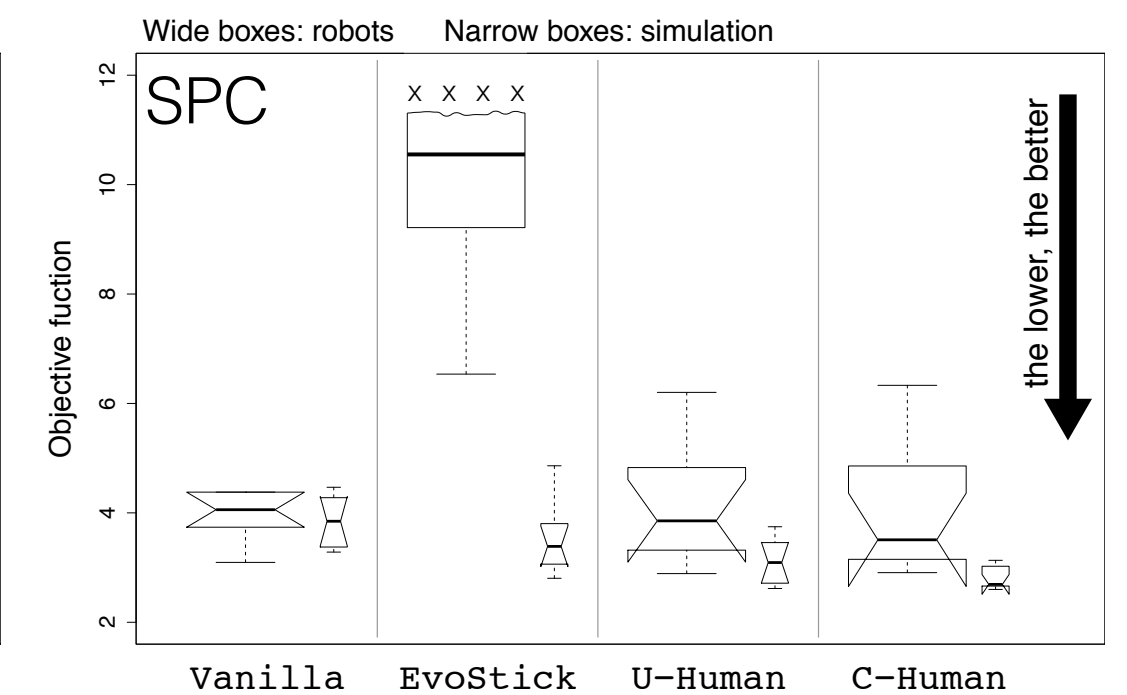
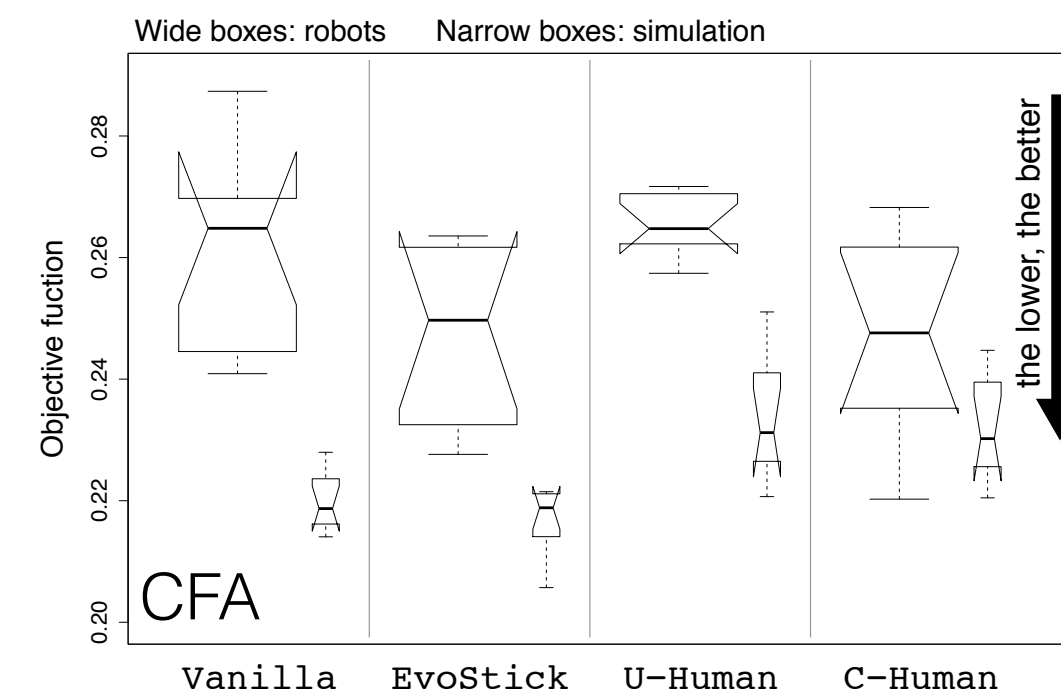
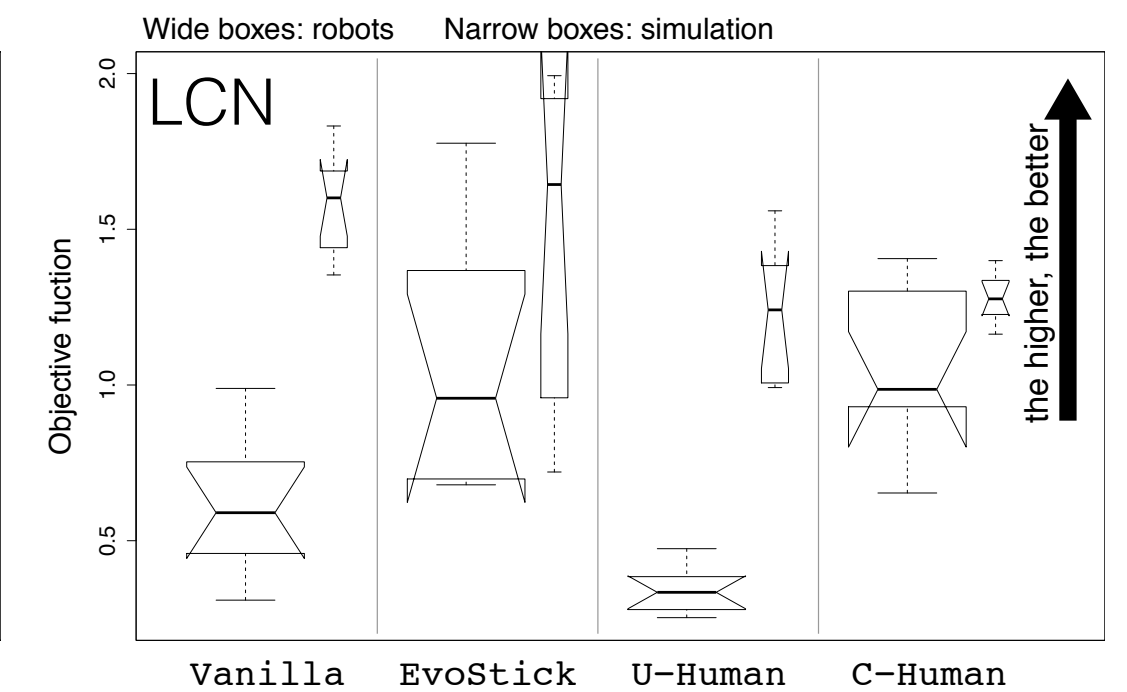
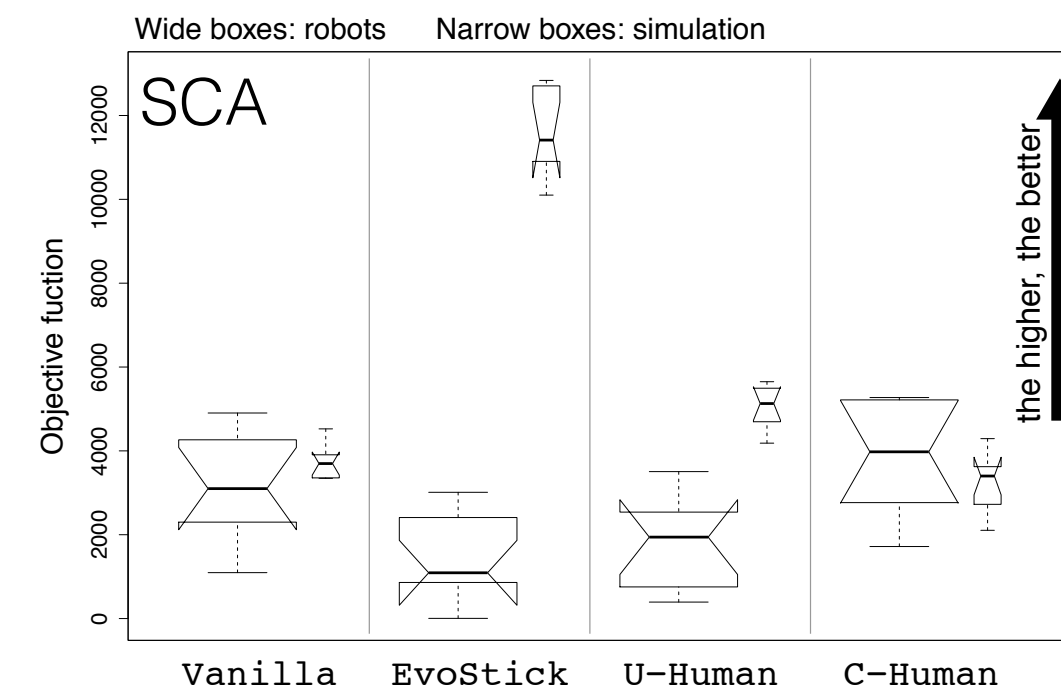
- Five **experts** proposed five problems
- Each expert propose/programs a solution in simulation
 - **unconstrained**
 - **constrained** to AutoMoDe's design space
- Automatic design methods have a fixed **computation budget**



SCA: shelter with constrained access
LCN: largest covering network
CFA: coverage with forbidden areas
SPC: surface and perimeter coverage
AAC: aggregation with ambient cues

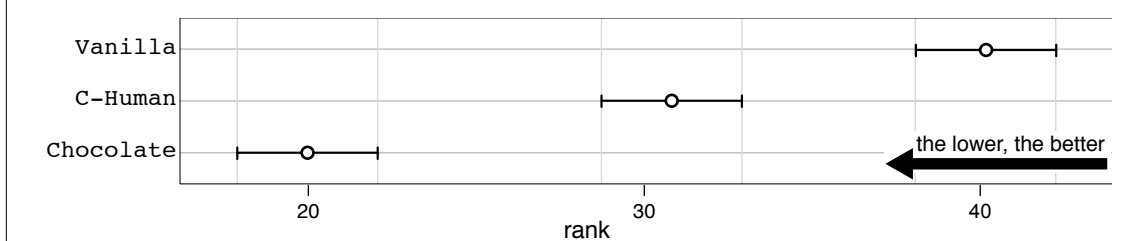
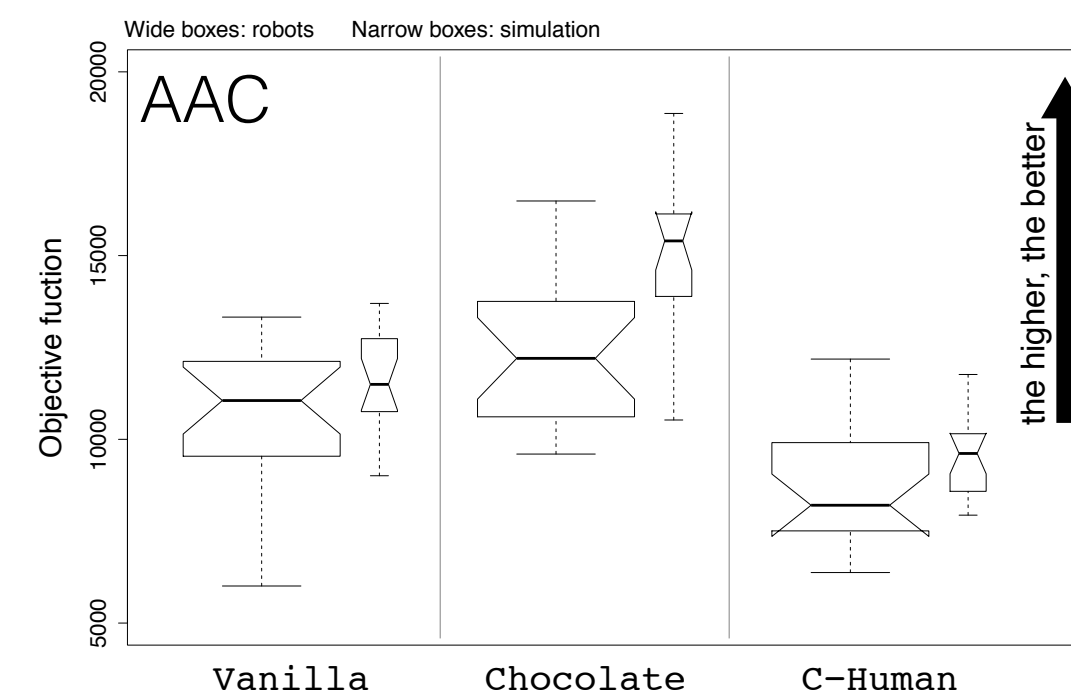
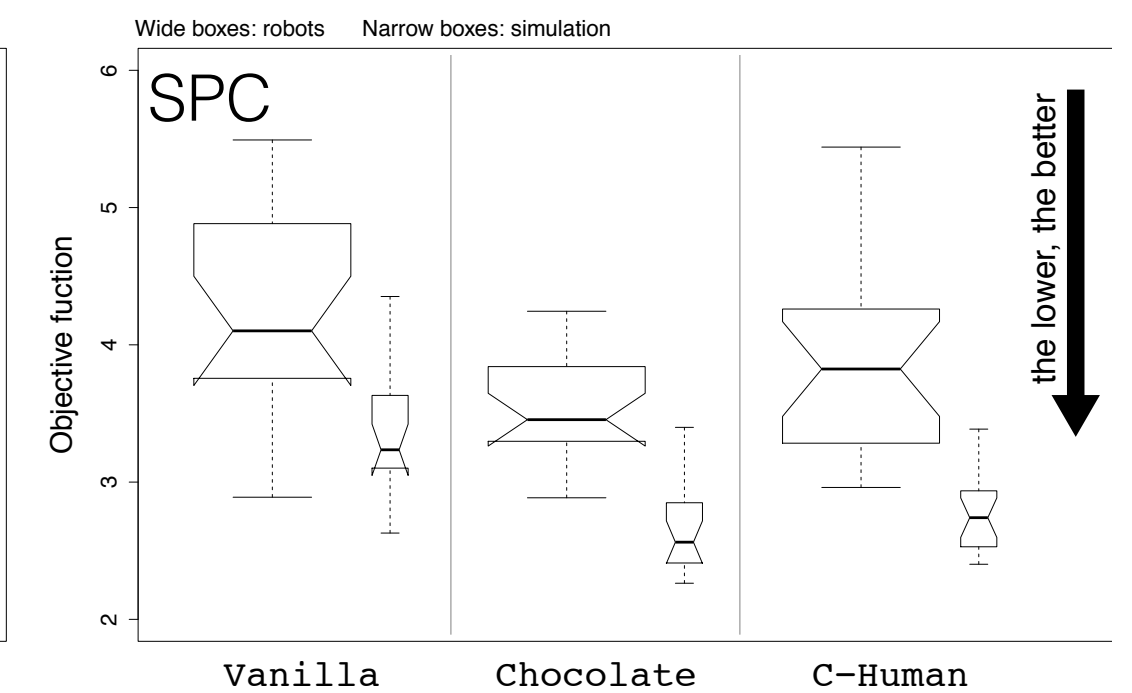
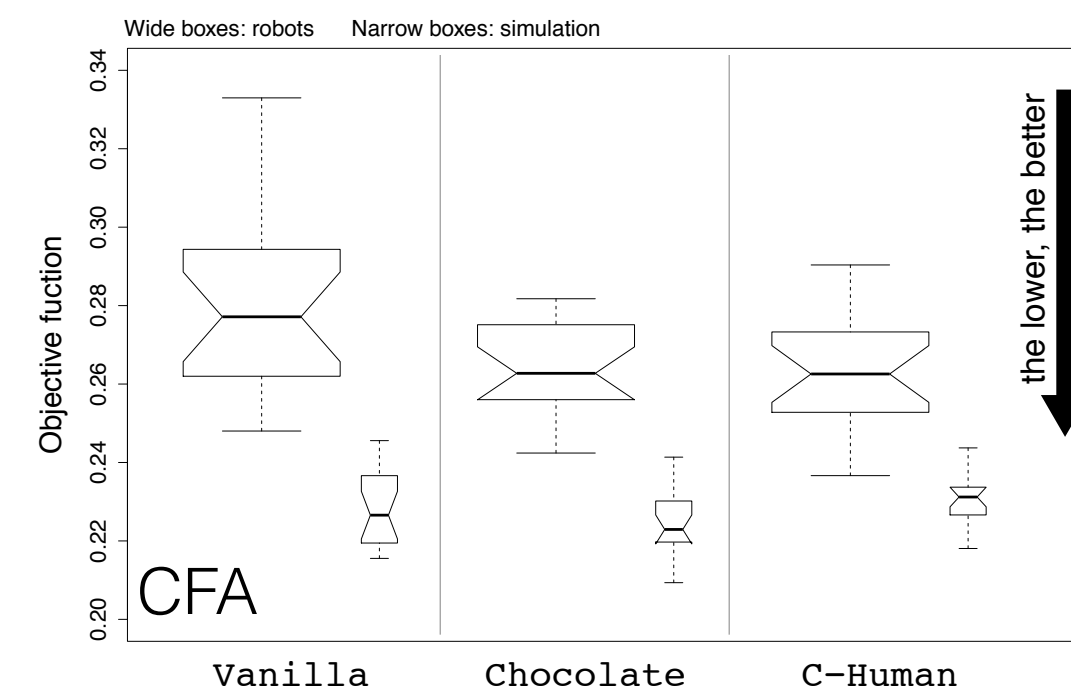
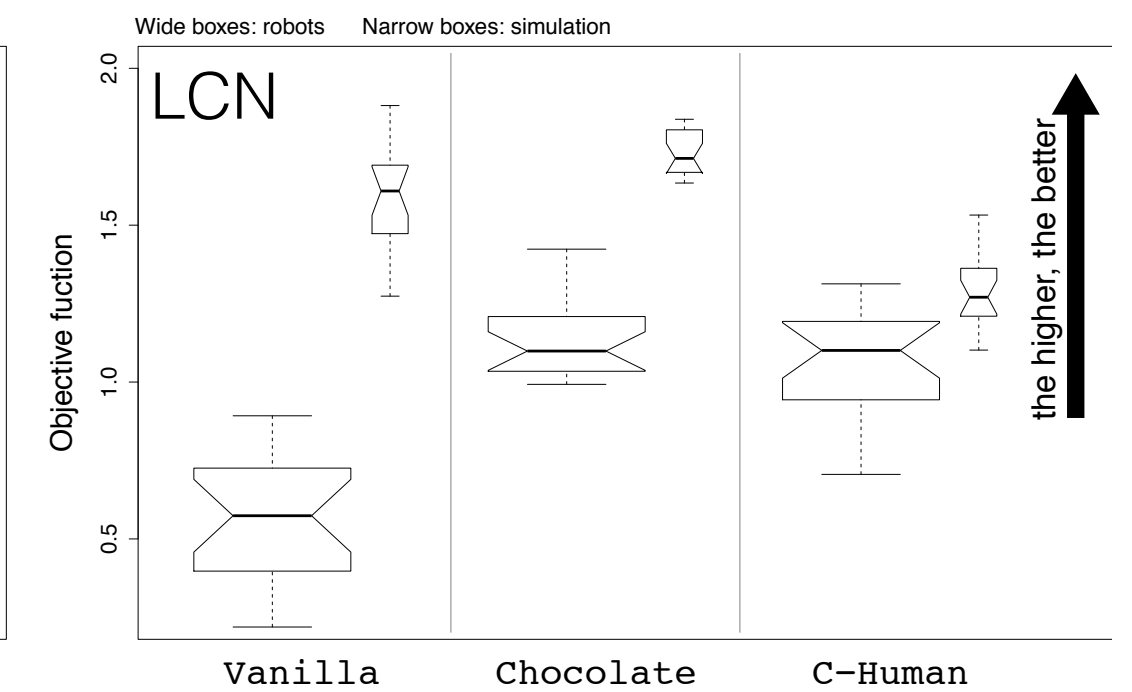
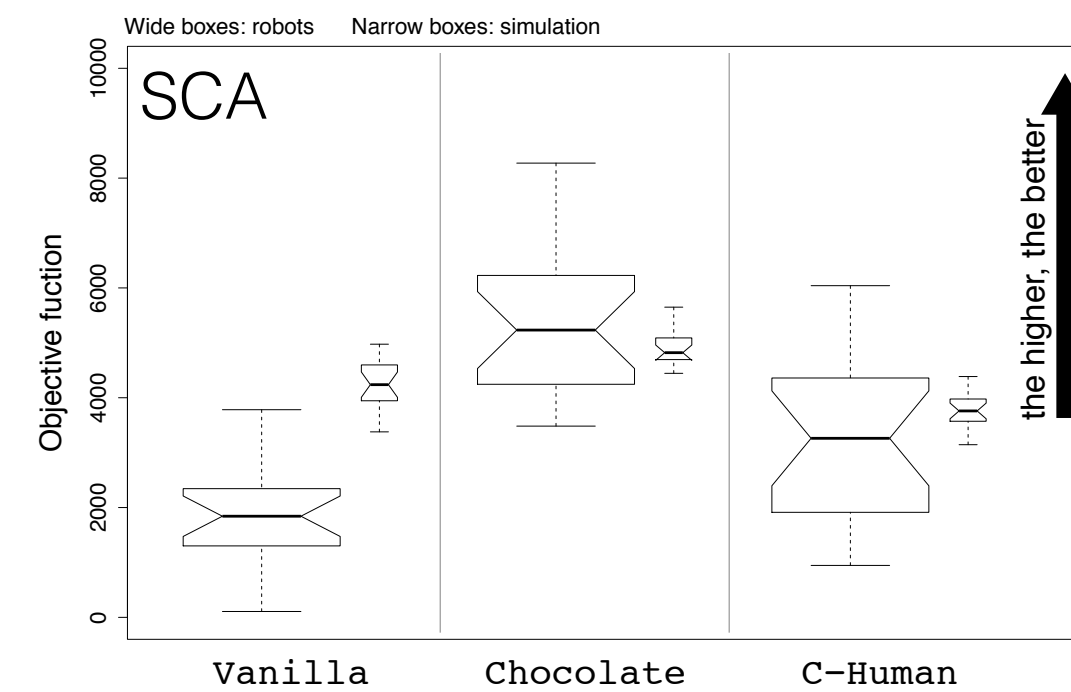
Results

- Performance metrics (SOO) defined by experts themselves
- Performance is compared in **simulation** and with **physical robots**



Results

- Performance metrics (SOO) defined by experts themselves
- Performance is compared in **simulation** and with **physical robots**
- An improved optimisation method for AutoMoDe leads to better results

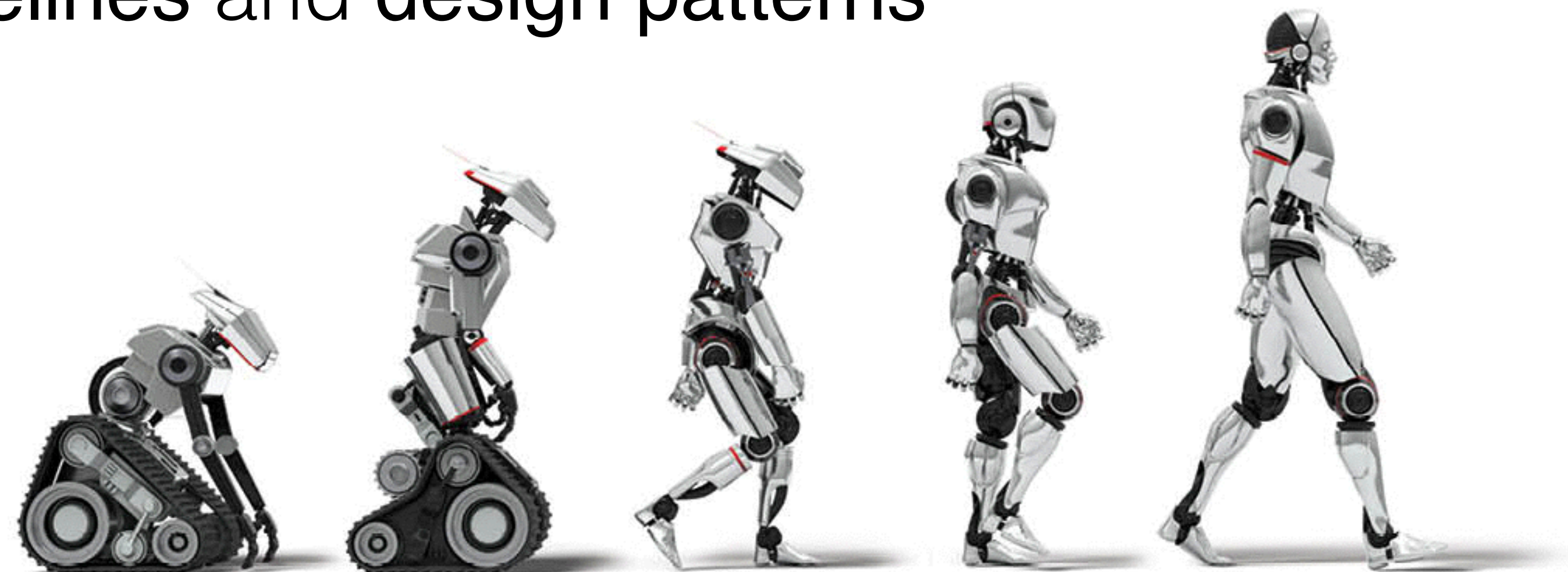


Take-Home Message #3

- Better generalisation can be obtained by introducing some bias
- Evolving modular solutions is a suitable approach for robot swarms
- Indirect encodings can have advantages if they exploit modules

Wrap Up

- Design for evolution of collective behaviours is often **counter-intuitive**
- To make evolution an intelligent design method, behave like the '**deus ex machina**'
- Collect studies to provide **guidelines** and **design patterns**
- Test and compare different approaches



Thanks for your attention

