from AutoMoDe to the Demiurge

IRIDIA's recent and forthcoming research on the automatic design of robot swarms

Swarm Intelligence: INFO-H-414

Mauro Birattari

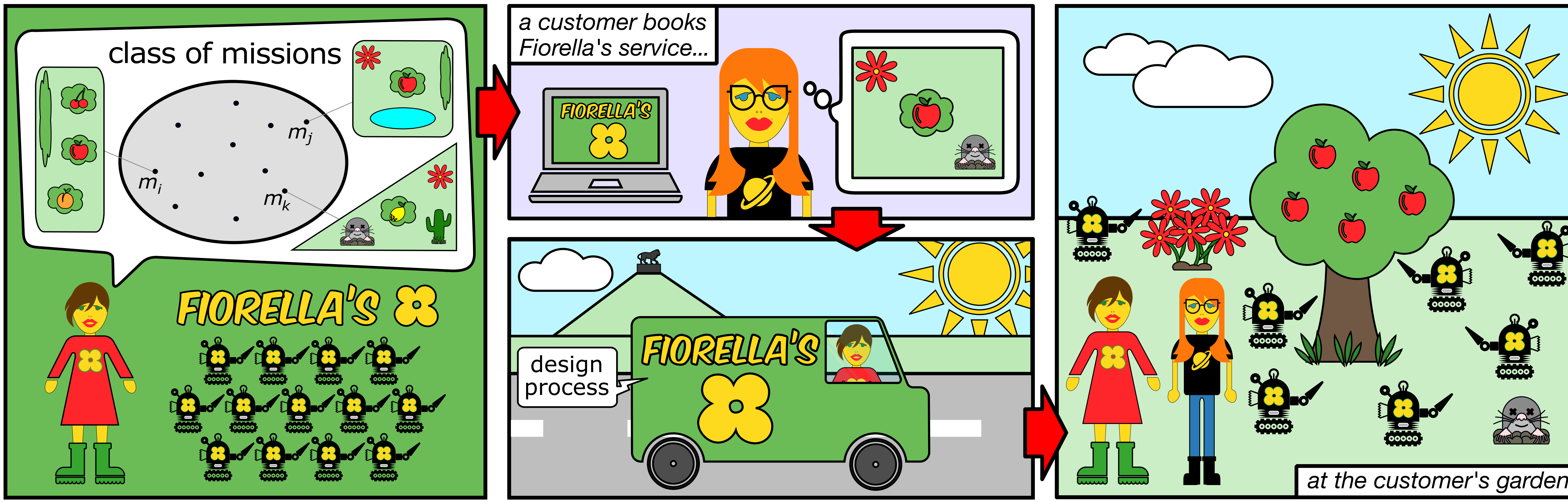
IRIDIA, Université libre de Bruxelles

European Research Council
Established by the European Commission
Supporting top researchers from anywhere in the world

European Commission

Horizon 2020
European Union funding for Research & Innovation

Fiorella’s swarm gardening



- Fiorella owns and operates a robot-swarm gardening business
- to provide the best service possible—but also to cut costs and maximize benefits—Fiorella uses automatic design to customize interventions
- behaviors are design while Fiorella drives her swarm to the customer’s
- due to time and cost constraints, Fiorella cannot intervene in the design process: once she reaches the customer, robots must be operational

outline

- introductory concepts
- definitions
- working hypothesis and idea
- from classical ER to AutoMoDe
- AutoMoDe-**`vanilla`** and AutoMoDe-**`chocolate`**
- towards the Demiurge

swarm robotics

in swarm robotics a mission is entrusted to a large group of robots

- a swarm is autonomous and self-organized:
  - no leader robot or external infrastructures
- a swarm is highly redundant
  - *homogeneous* of *heterogeneous*… but nobody is indispensable
- robots are capable of local perception and communication only
  - each robot interacts with few neighbours and is unaffected by swarm size
- robots operate in parallel on multiple tasks
  - they switch from task to task in an autonomous and self-organized way

autonomy, self-organization, redundancy, locality, and parallel execution promote fault tolerance, scalability, and flexibility

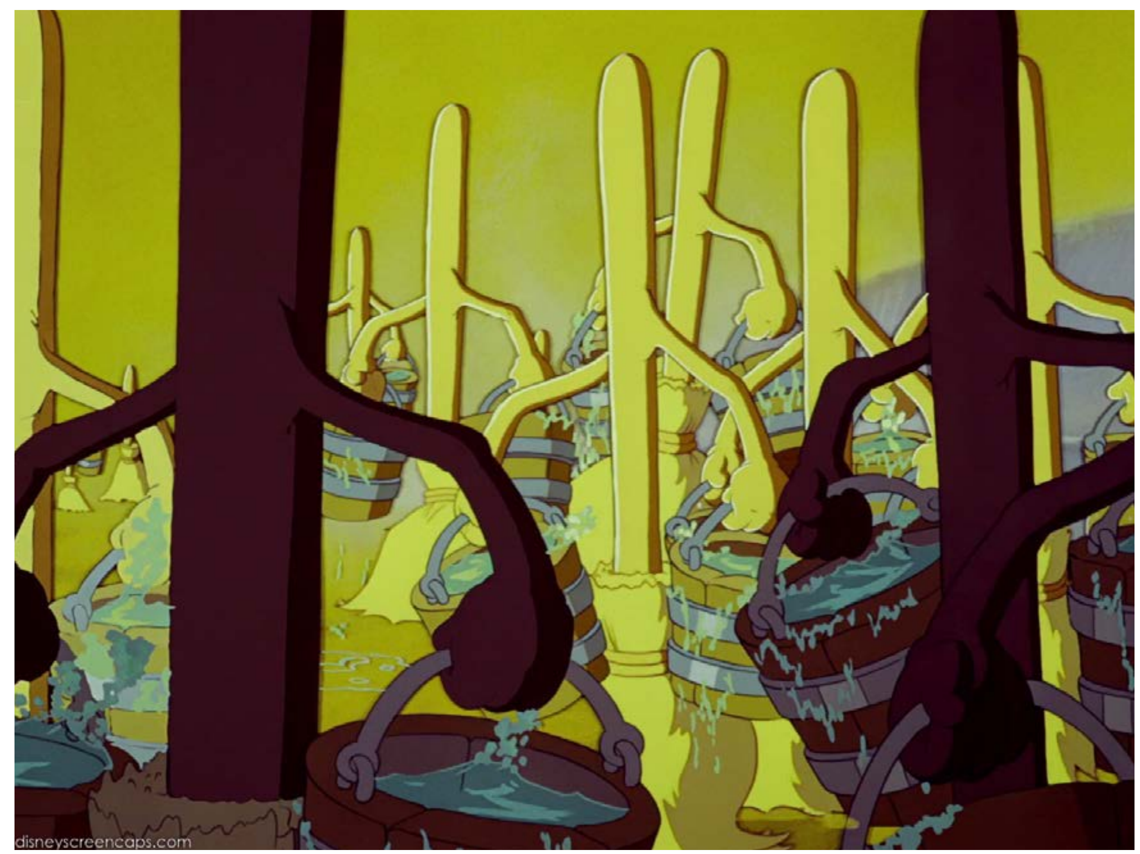
state of the art and limitations

- ☑ self-organization is viable
- ☒ no general design methodology
- ☒ swarms are mostly designed by hand via trial and error
  - high costs
  - not predictable/repeatable
  - no guarantees

the lack of an engineering methodology prevents real-world applications

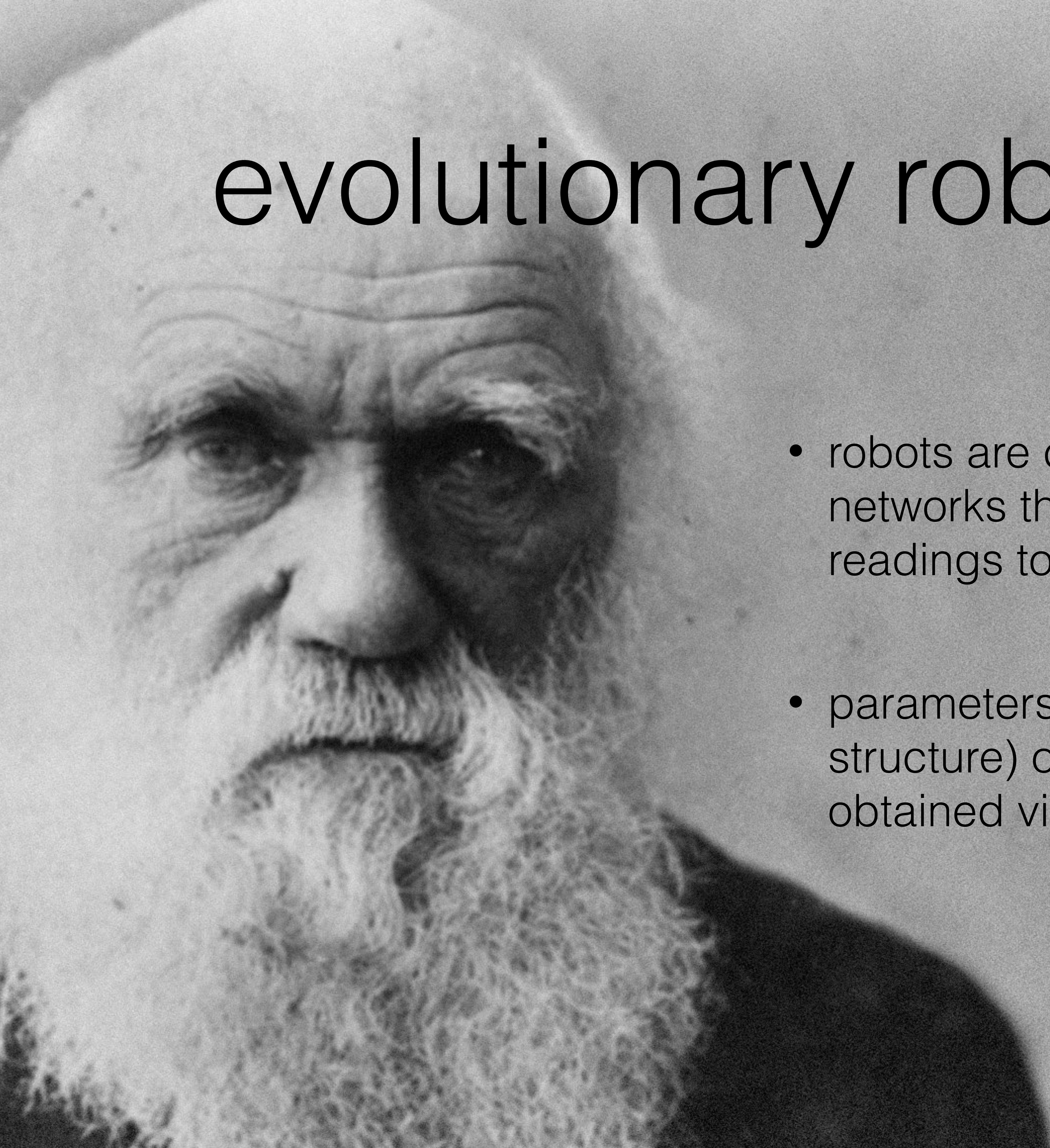
the design challenge in swarm robotics

design the individual …



to obtain desired swarm-level properties

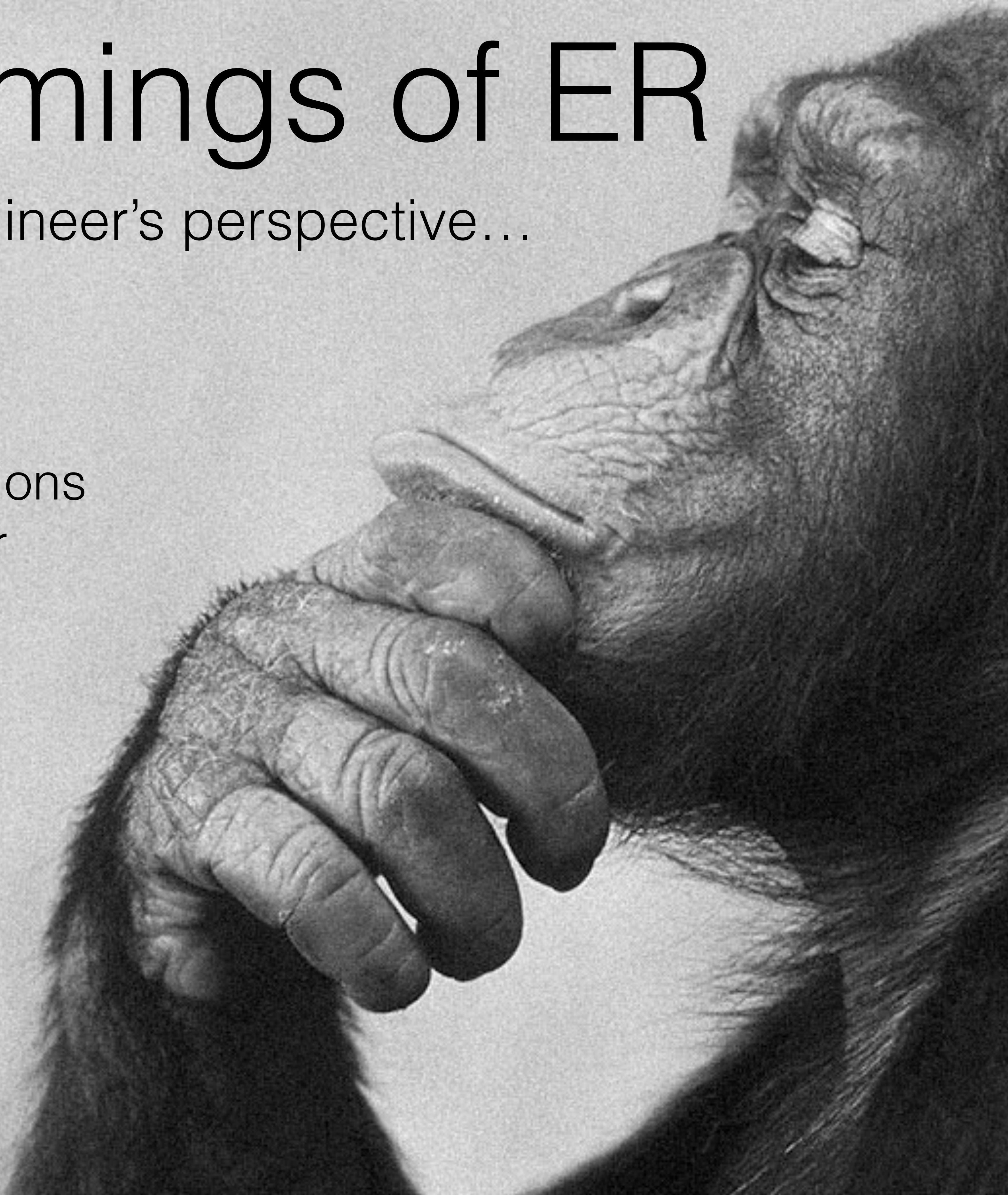
evolutionary robotics



- robots are controlled by neural networks that map sensor readings to control actions
- parameters (and possibly structure) of neural network are obtained via evolution

shortcomings of ER

from an engineer's perspective…



- focus on research questions relevant to biology rather than engineering
- unclear methodology concerning role of experimenter

offline
automatic design

design phase

in simulation

operation phase

in target environment

vs

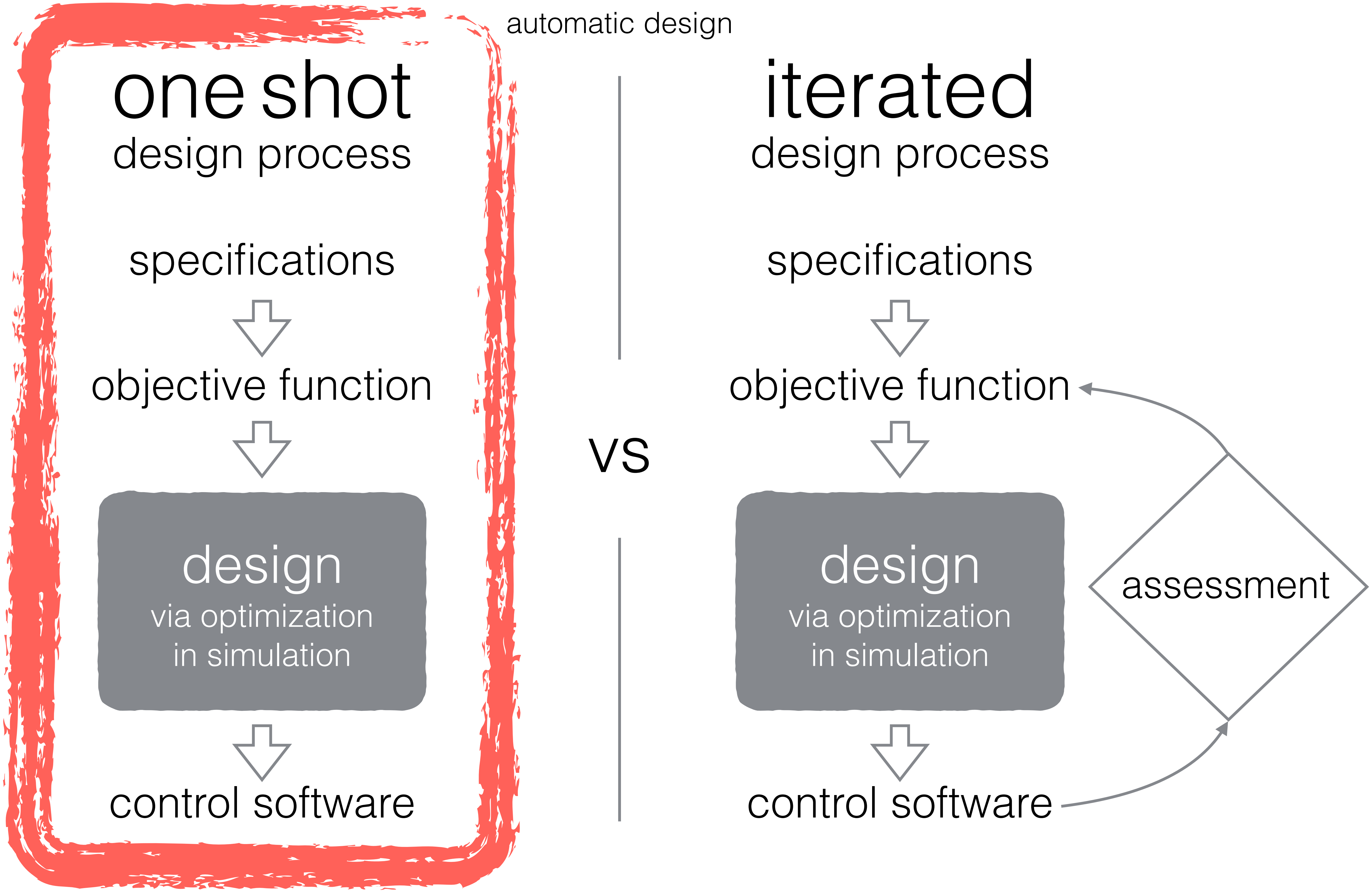
online
automatic design

continuous design during operation

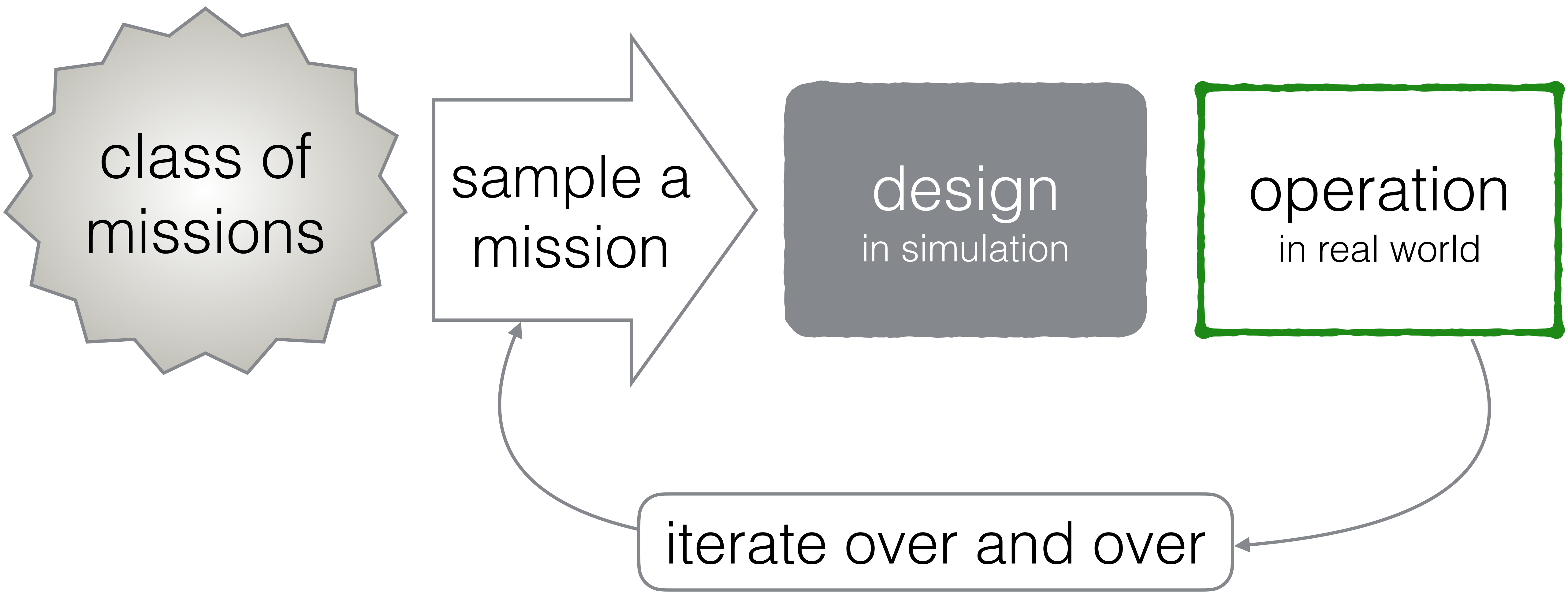
in target environment

not an exhaustive taxonomy: hybrids and variants are possible

offline
automatic design



overall problem model



working hypothesis and idea

offline

automatic design

design

in simulation

reality gap

operation

in real world

whp

~

machine learning

training set

generalization

test set

idea: handle reality gap as a generalization problem

a bias/variance dilemma?

a possible explanation of the inability to successfully cross the reality gap shown by some/many robot swarms obtained via classical ER:

- NN could be too *powerful:* low bias / high variance
- they could *overfit* idiosyncrasies of simulator

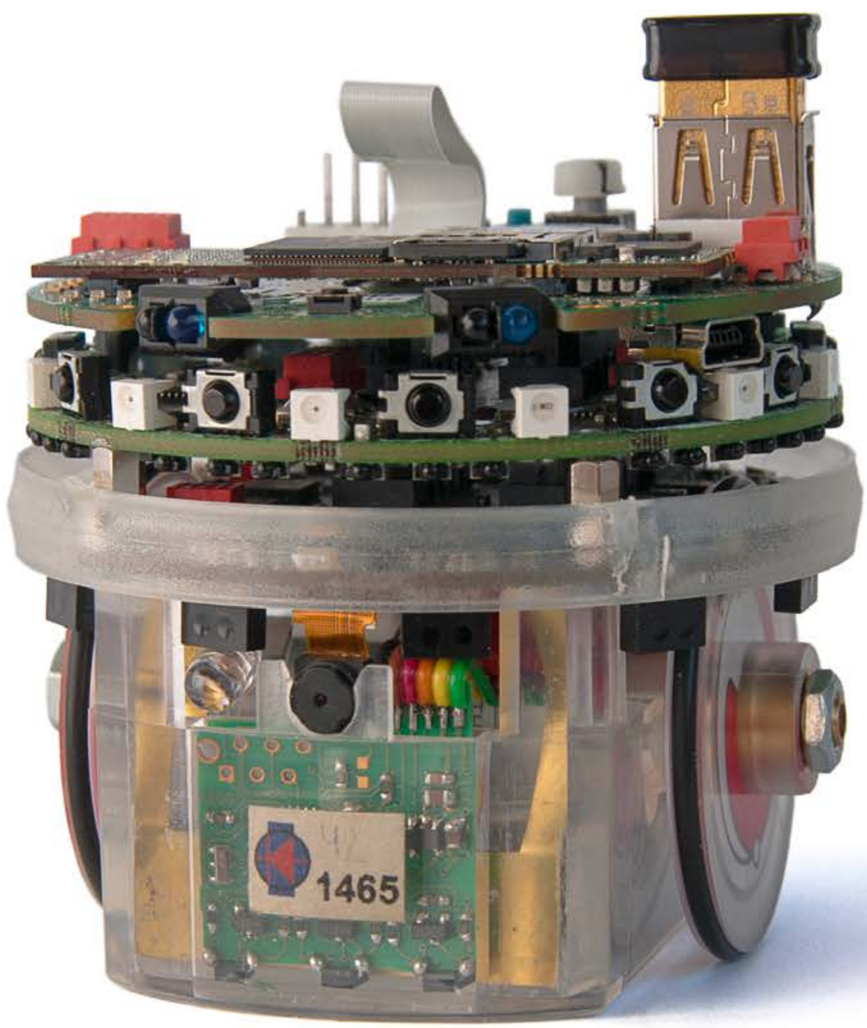
AutoMoDe

- limit power of control software by injecting bias
- finite state machines obtained by assembling and fine-tuning preexisting parametric modules
- modules are *mission independent* :
  defined *a priori* on the basis of robot capabilities
  (an example follows…)

AutoMoDe–vanilla

RM1: a reference model of the e-puck

(subset of sensors/actuators)



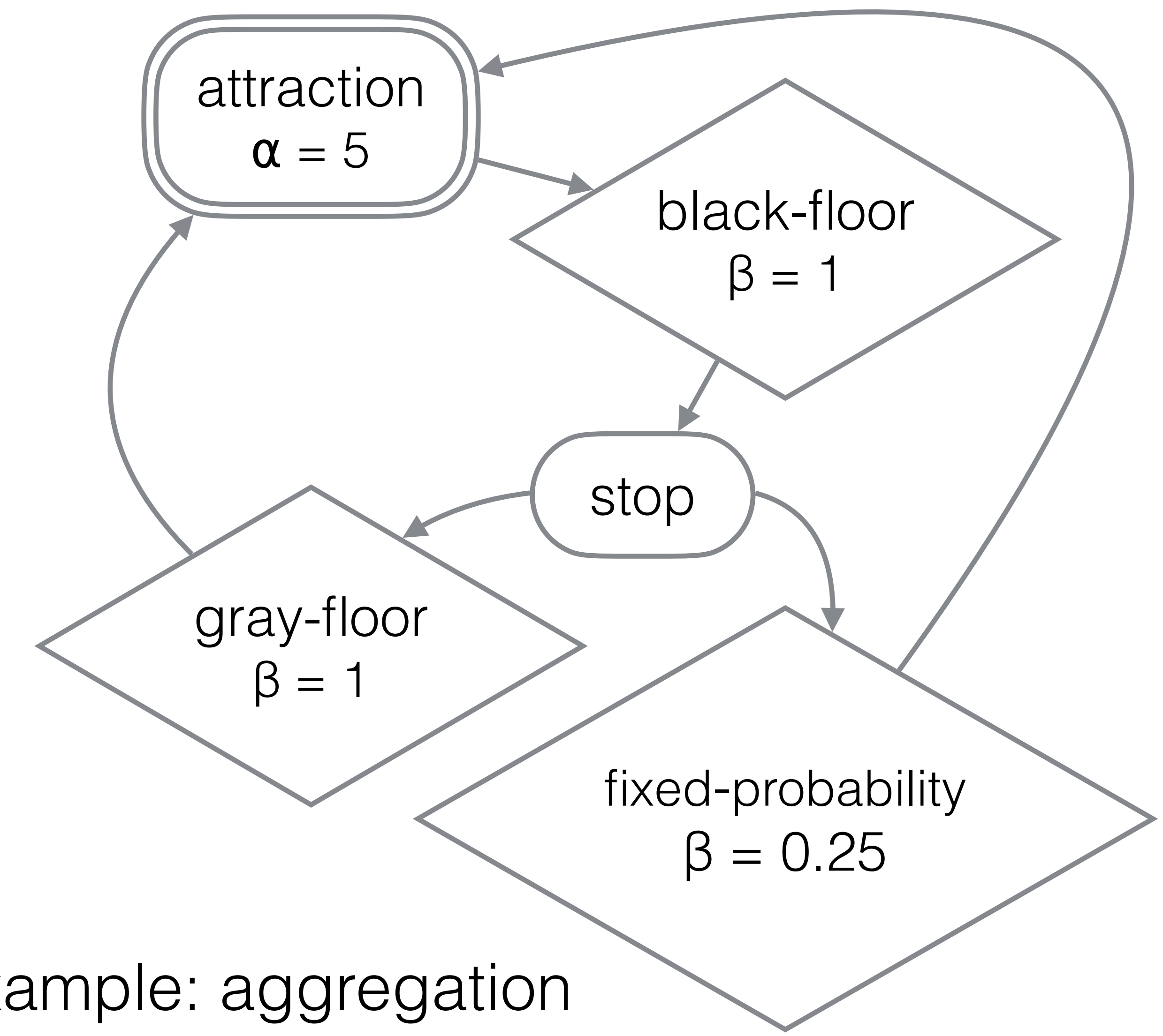
sensors/actuators	variables
proximity	$prox_i \in [0,1]$ & $\angle q_i$ with $i \in \{1,2,\ldots,8\}$
light	$light_i \in [0,1]$ & $\angle q_i$ with $i \in \{1,2,\ldots,8\}$
ground	$gnd_i \in \{0,.5,1\}$ with $i \in \{1,2,3\}$
range and bearing	$n \in \mathbb{N}$, r_m, & $\angle b_m$ with $m \in \{1,2,\ldots,n\}$
wheels	v_l & $v_r \in [-V,+V]$ with V = 0.16 m/s

period of control cycle: 100 ms

AutoMoDe–`vanilla`

parametric behaviors & transition conditions

behaviors	conditions
exploration	black-floor
stop	gray-floor
phototaxis	white-floor
anti-phototaxis	neighbor-count
attraction	inverted-neighbor-count
repulsion	fixed-probability

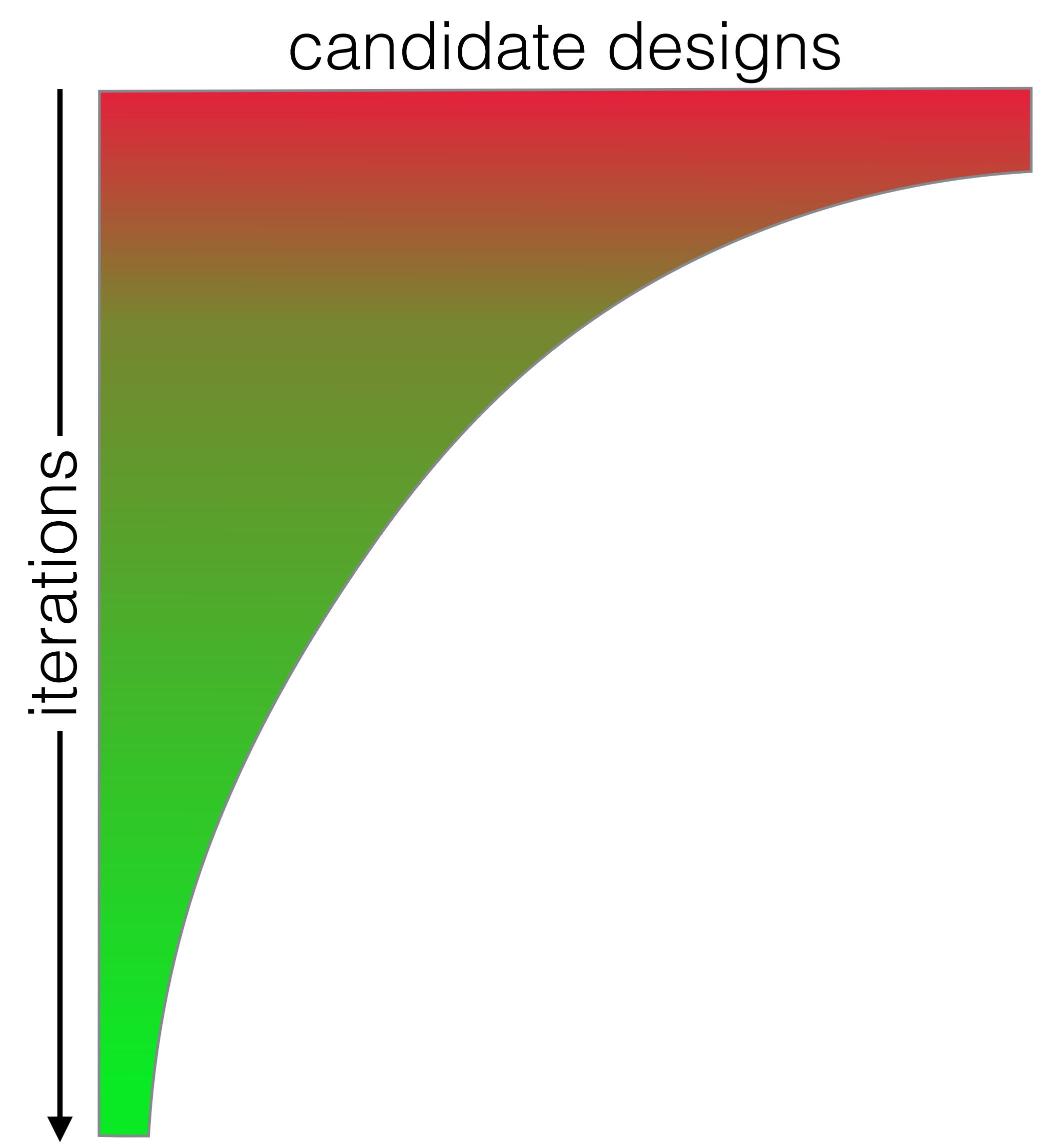


example: aggregation

AutoMoDe–`vanilla`

optimization algorithm: F-Race (Birattari *et al.*, 2002)

- sample candidate designs
- iteratively evaluate them
- drop dominated ones
- stop when one survives or evaluation budget depletes



experiments

- design in simulation with ARGoS
- tests with swarm of 20 e-puck robots

- two missions: *aggregation* and *foraging*

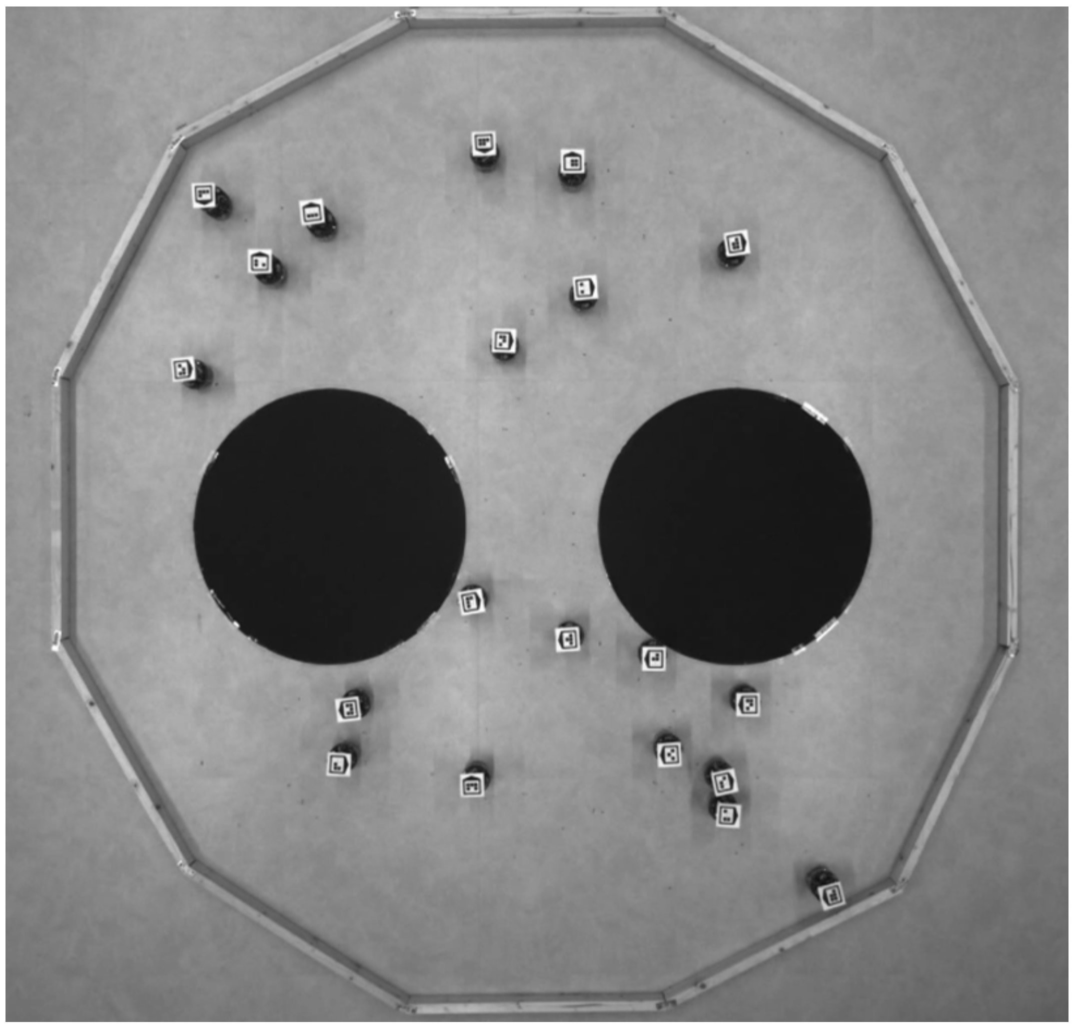
- three budget levels: 10k, 50k, and 200k

- two design methods: `vanilla` and `EvoStick`

no modification whatsoever allowed to adapt design methods to mission or budget level !!!

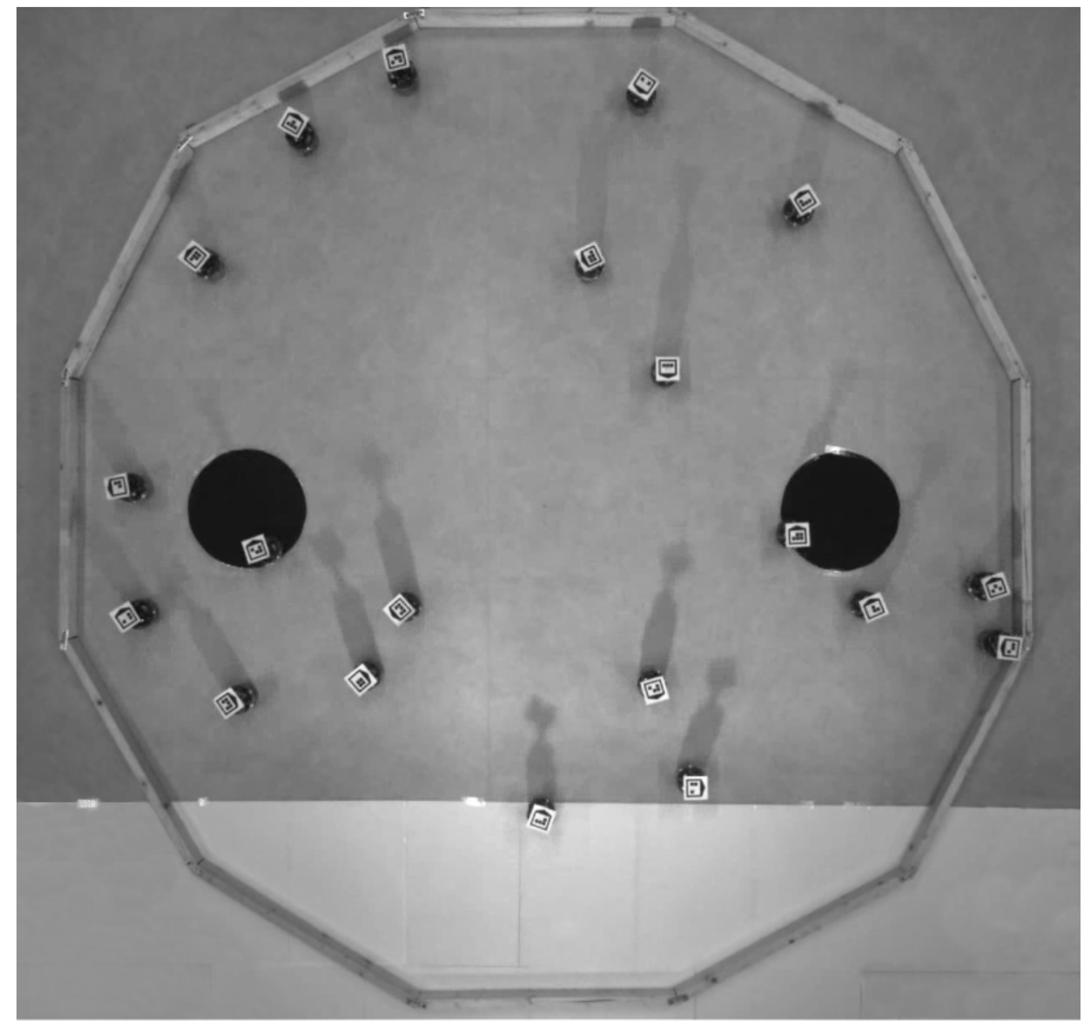
AutoMoDe-`vanilla`

aggregation



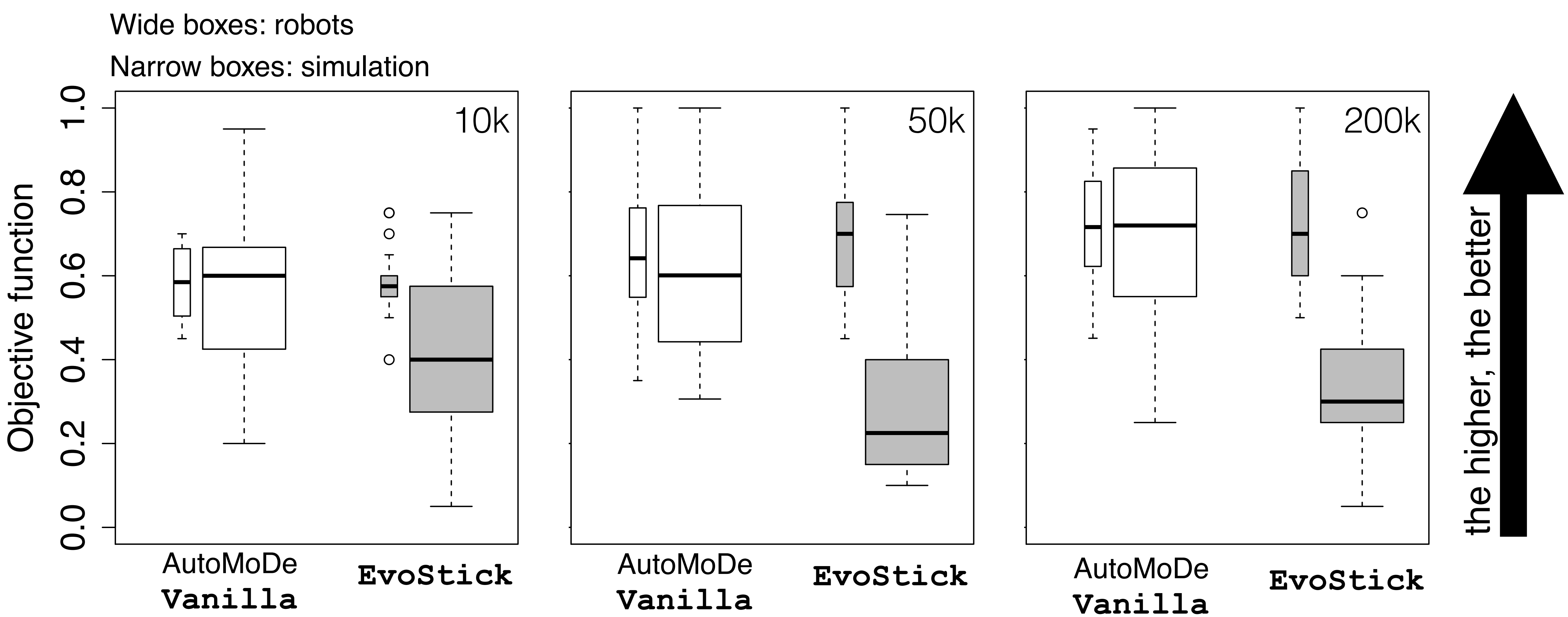
$F = \max(N_a, N_b)/N$

foraging



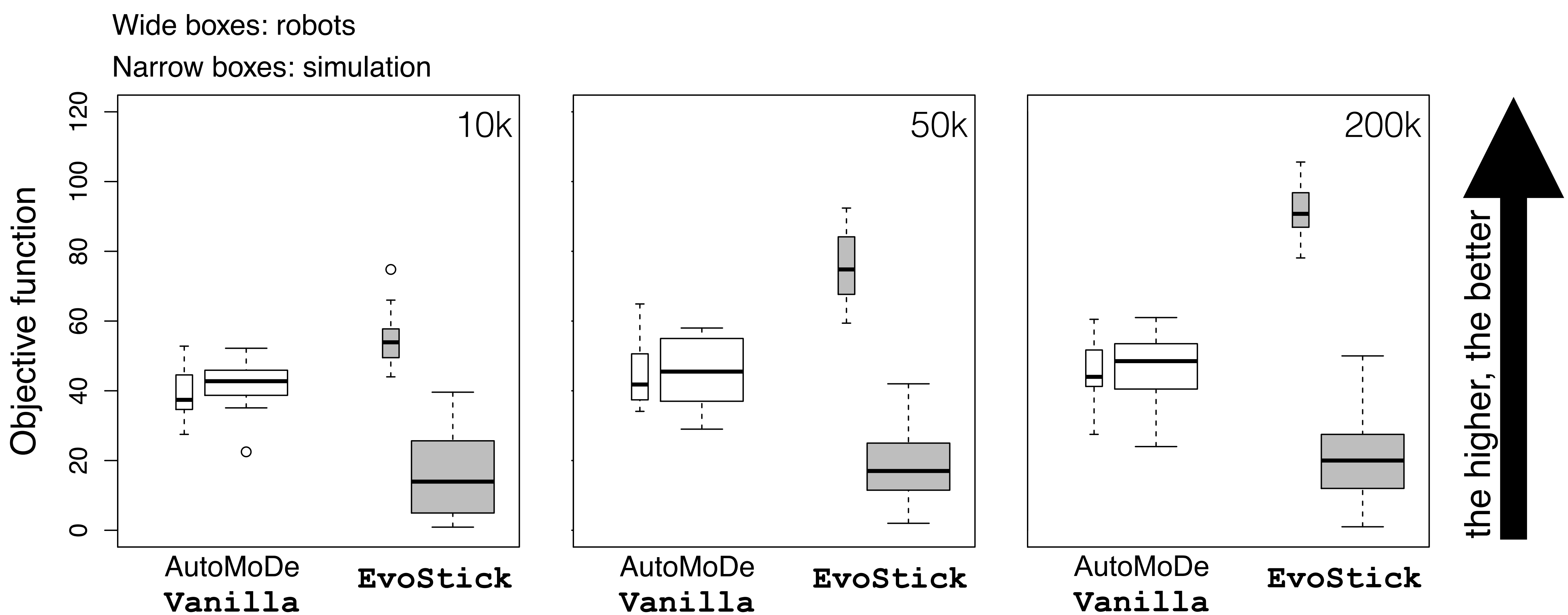
$F = N_o$

results: aggregation

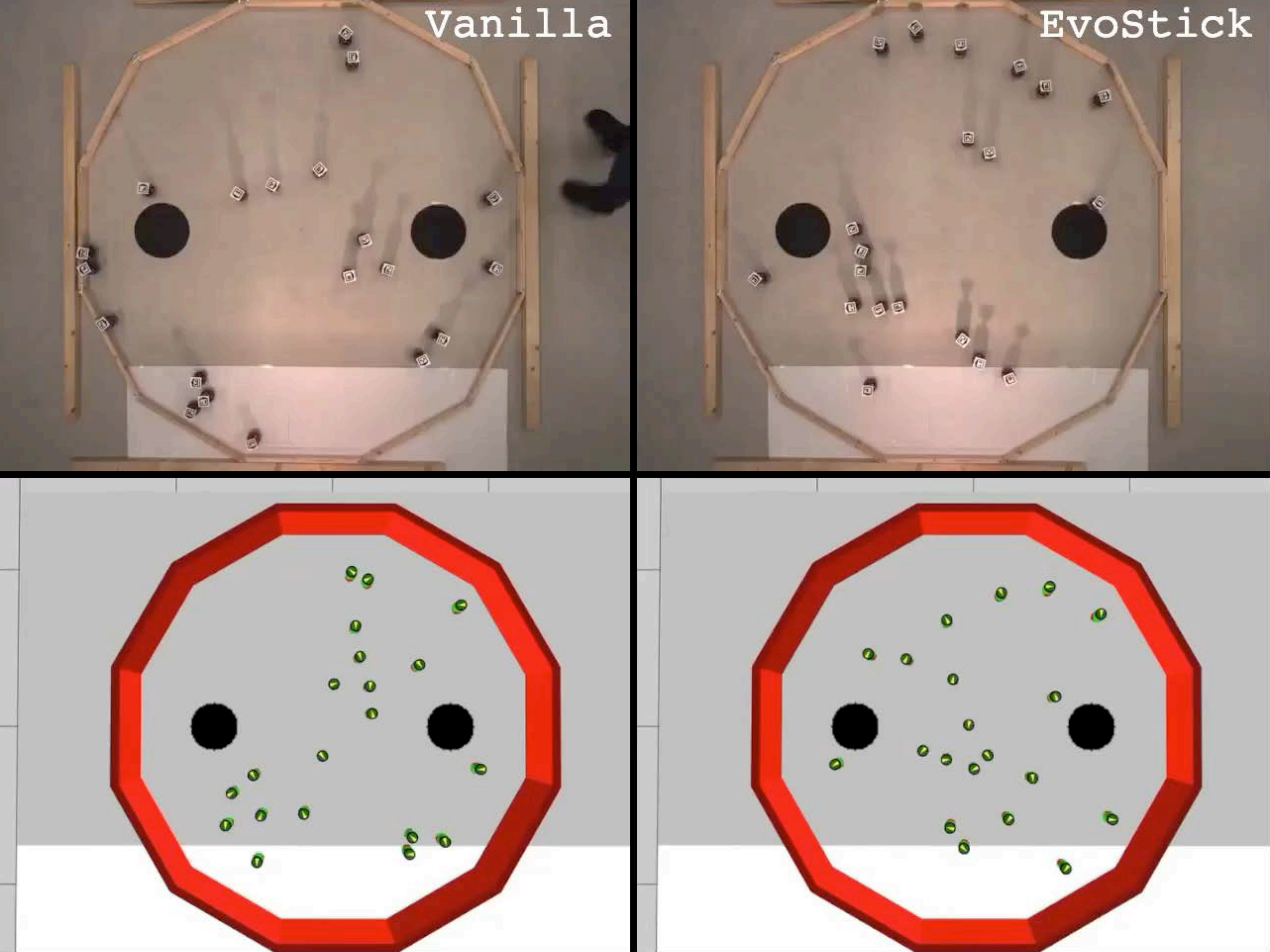


EvoStick's performance is good in simulation but drops in robot experiments

results: foraging



EvoStick is excellent in simulation but poor in robot experiments. The gap grows with the budget



Vanilla
EvoStick

remarks

- `vanilla`'s modules proved to be general enough to produce control software for two different missions
- whether they are general enough to produce control software for *any* mission that could be possibly accomplished by robots described by reference model RM1 is an *empirical question*

experiment with humans

5 human designers

- PhD students at IRIDIA in swarm robotics
- proficient in ARGoS and acquainted with the e-puck
- unaware of AutoMoDe-**`vanilla`** and its functioning

were asked to

- conceive a mission – compatibly with given reference model and with constraint on arena, possible obstacles, ground color, etc
- solve *another* mission by assembling **`vanilla`**'s modules
- solve *yet another* mission by direct C++ programming

experiments

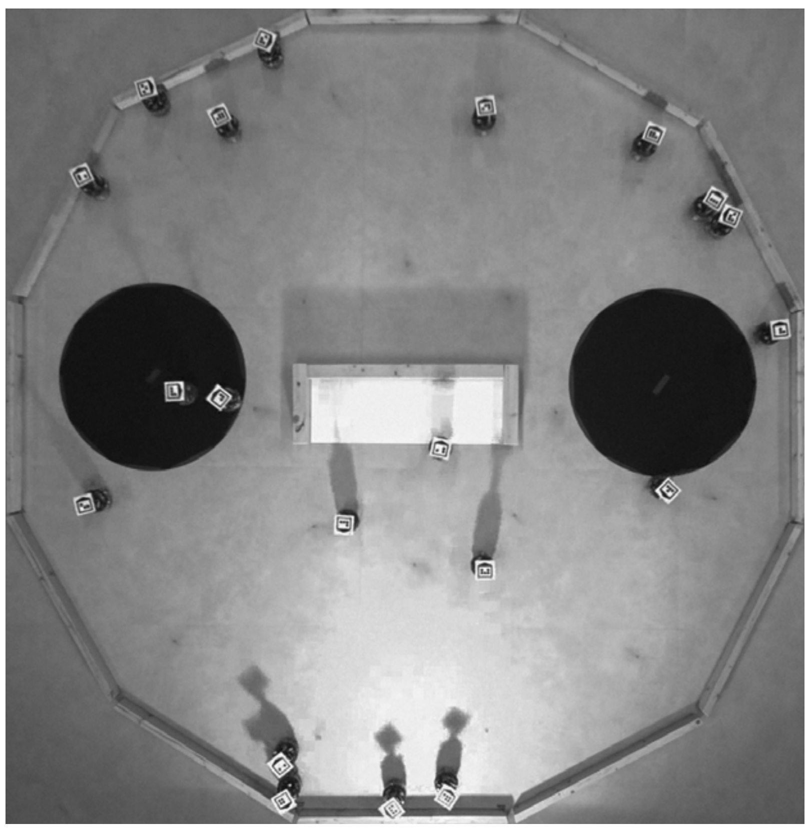
- design in simulation with ARGoS
- tests with swarm of 20 e-puck robots
- budget: 200k for automatic methods (~2h30) and 4h for humans

- 5 missions proposed by human designers: *shelter with constrained access, largest covering network, coverage with forbidden areas, surface and perimeter coverage, aggregation with ambient cues*

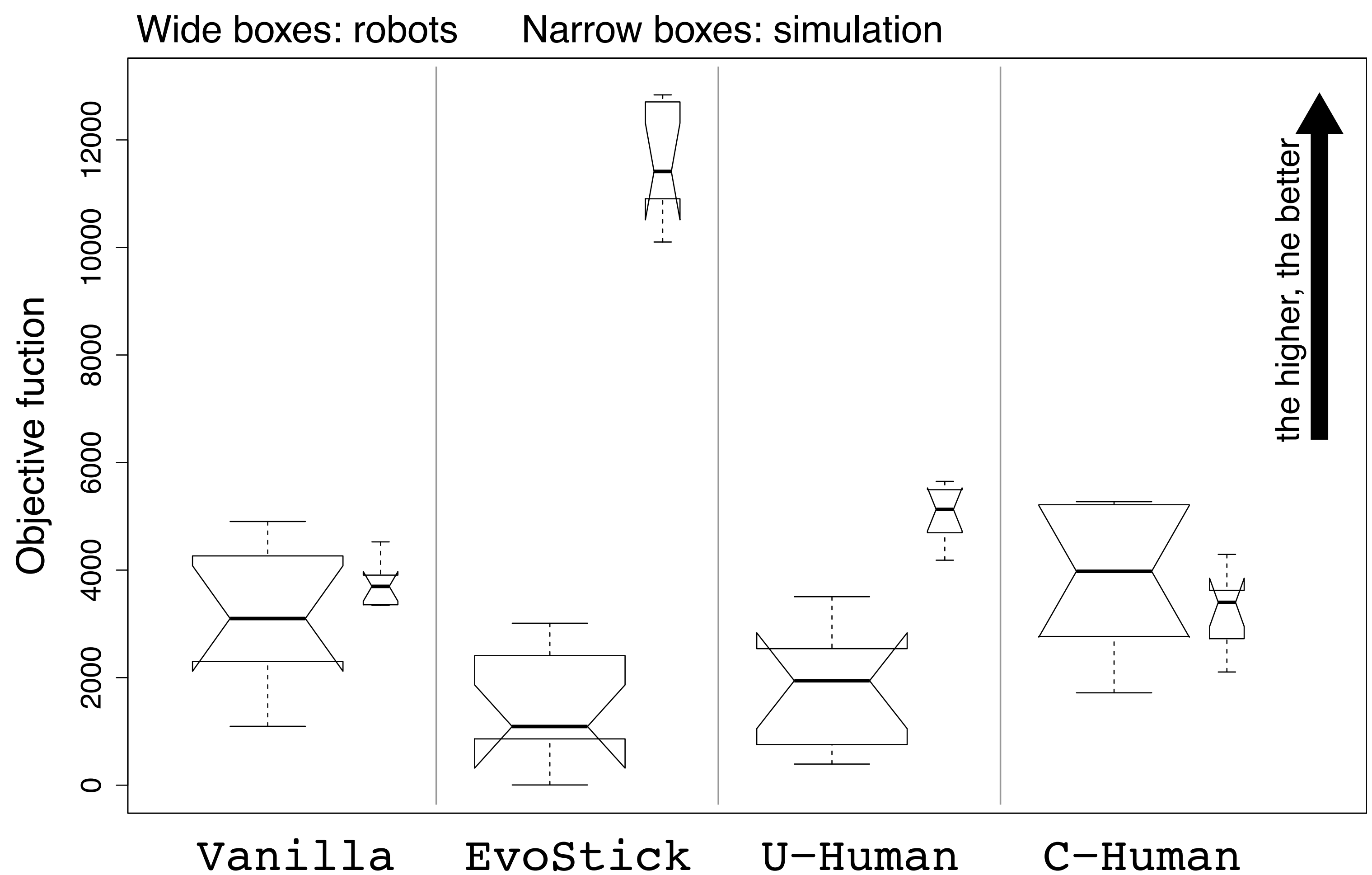
- 4 design methods: `vanilla`, `EvoStick`, `C-Human`, `U-Human`

no modification whatsoever allowed to adapt the two automatic design methods to new missions !!!

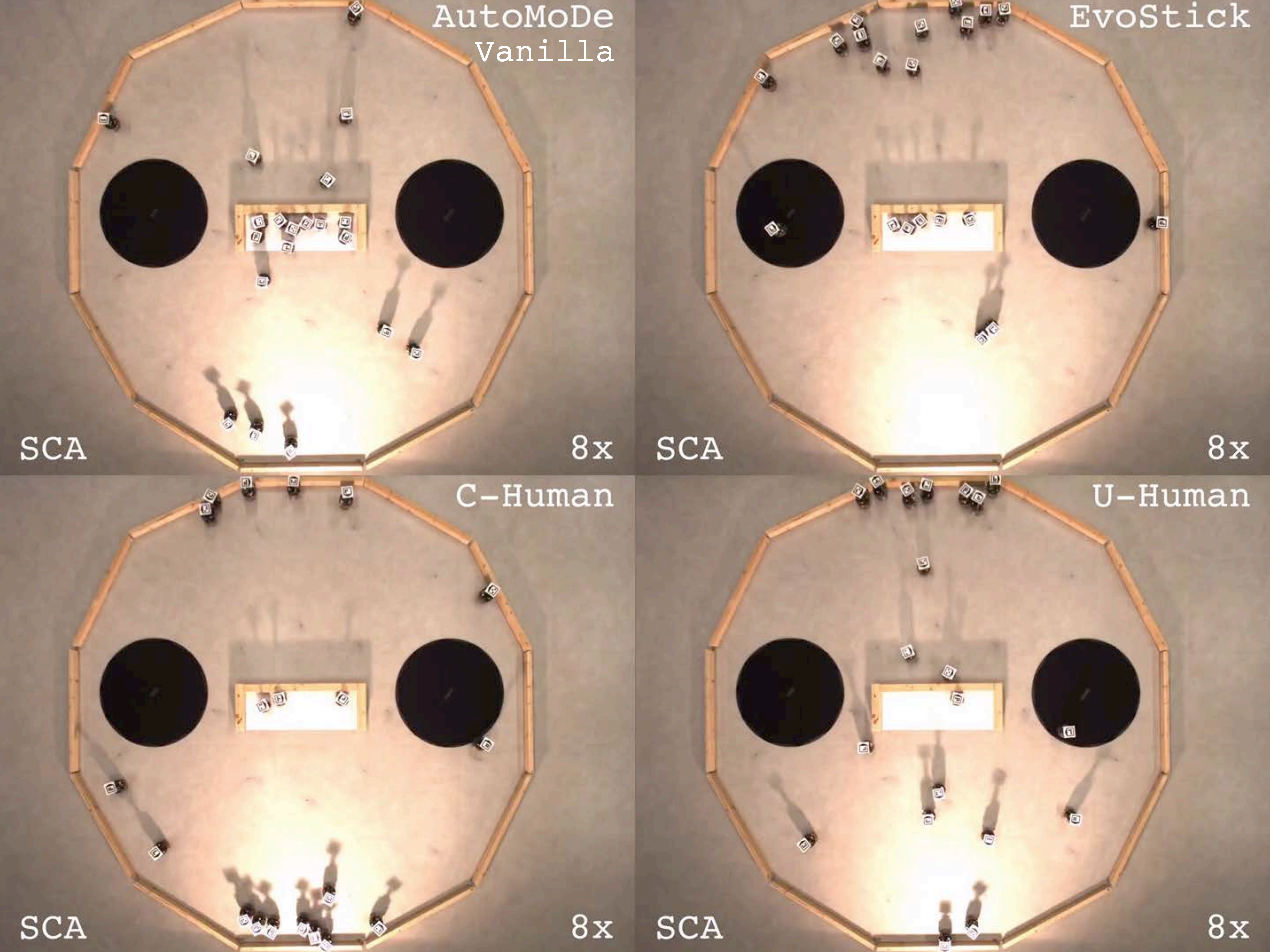
mission and results



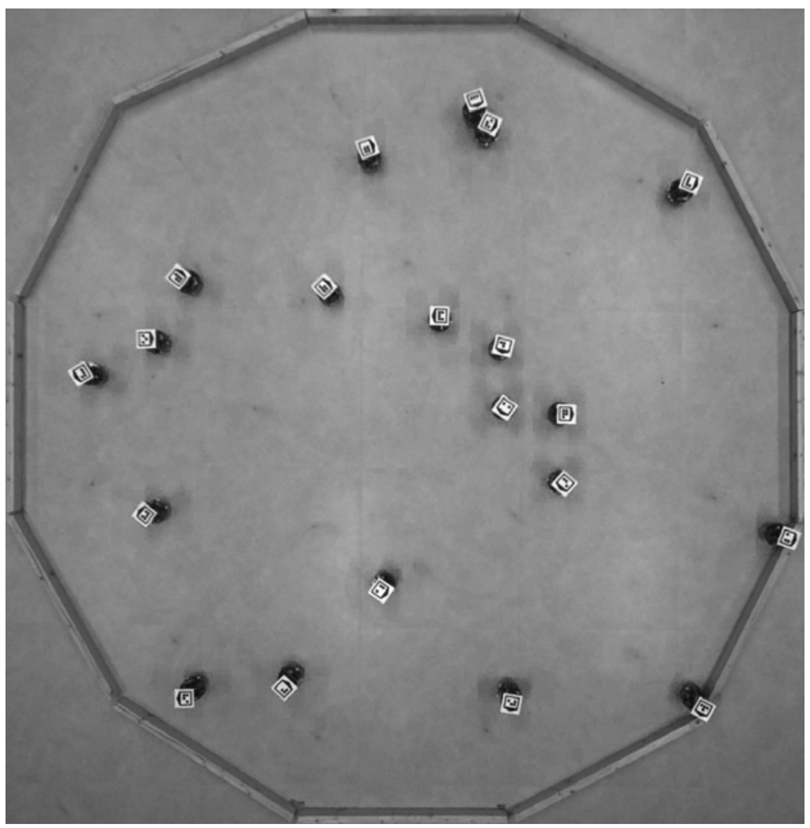
$F = \sum_{t=1}^{T} N(t)$



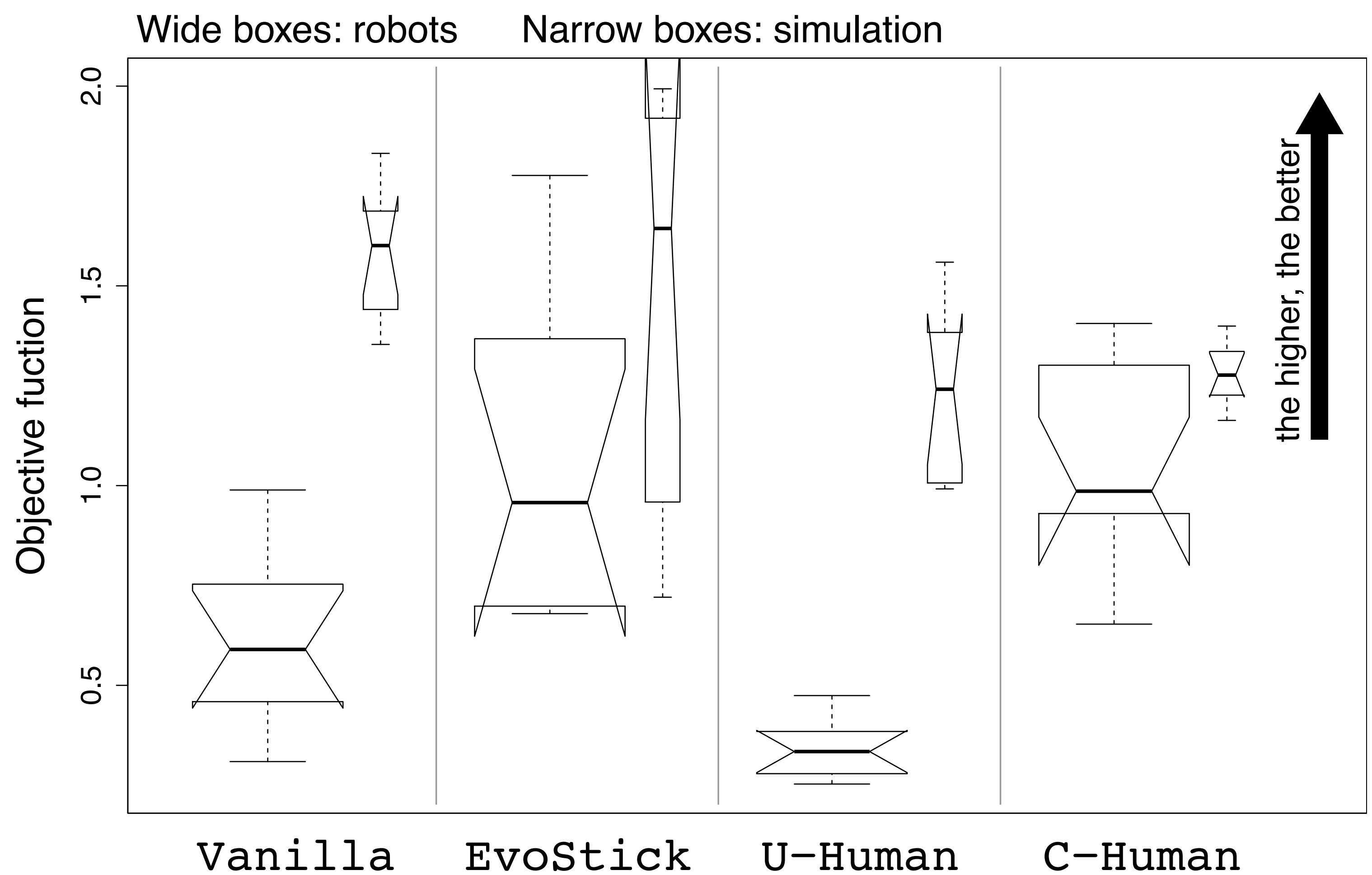
SCA – shelter with constrained access: robots should aggregate in the white shelter


AutoMoDe
Vanilla
SCA
8x
EvoStick
SCA
8x
C-Human
SCA
8x
U-Human
SCA
8x

mission and results

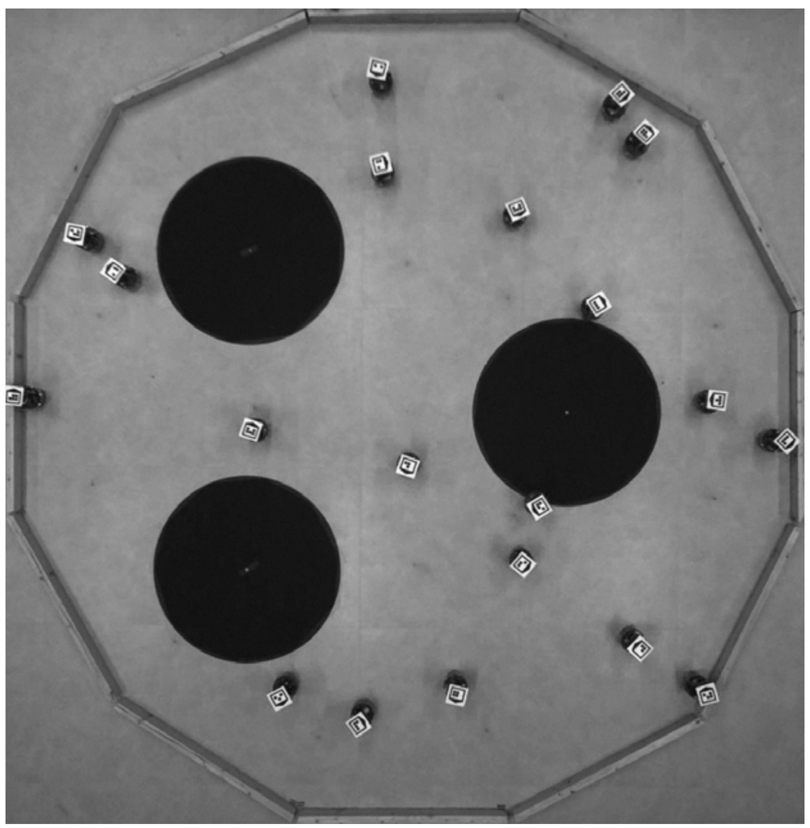


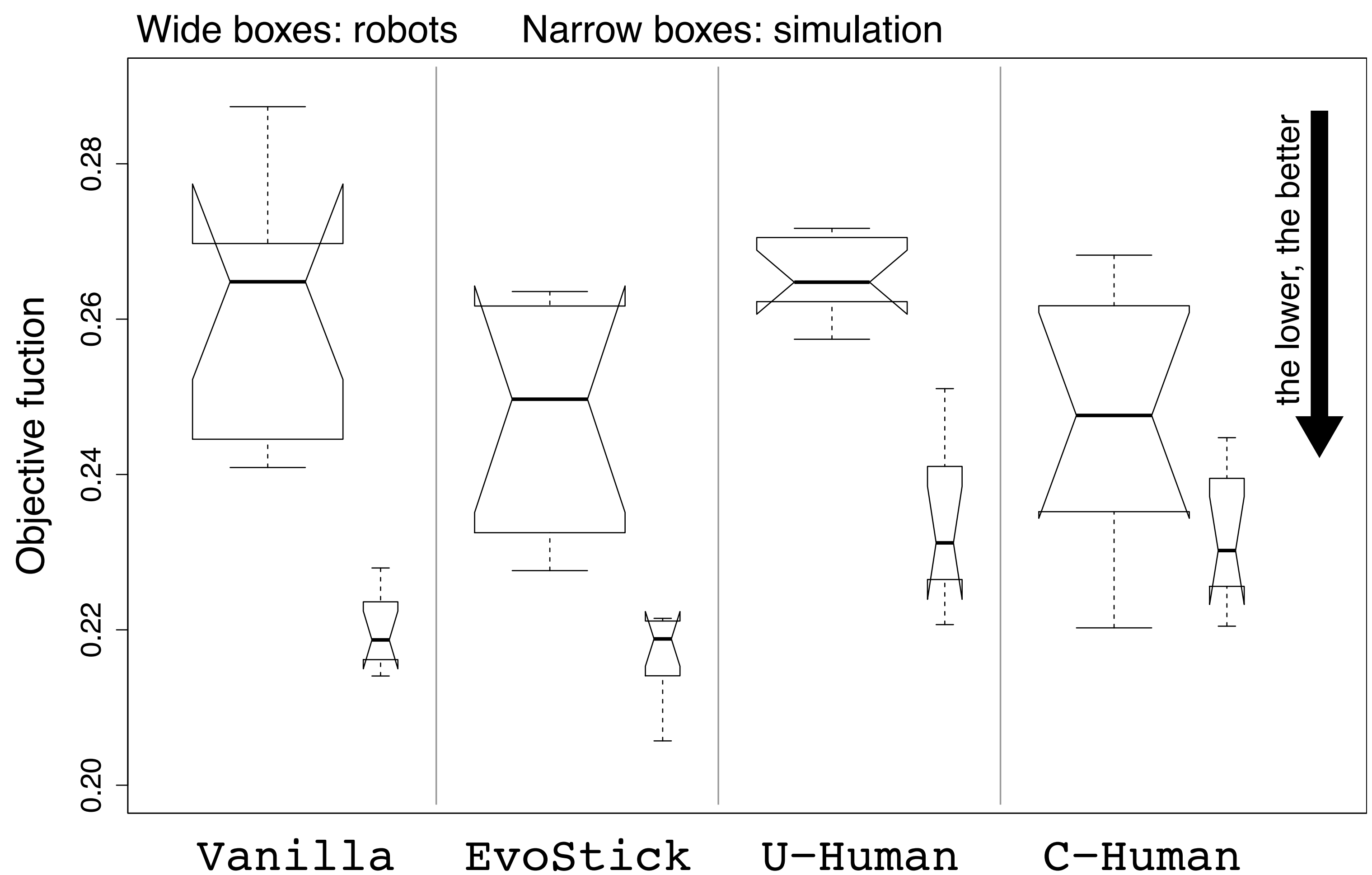
$F = A_{C(T)}$



LCN – largest covering network: robots should cover the largest possible area while maintaining connection with one another

mission and results

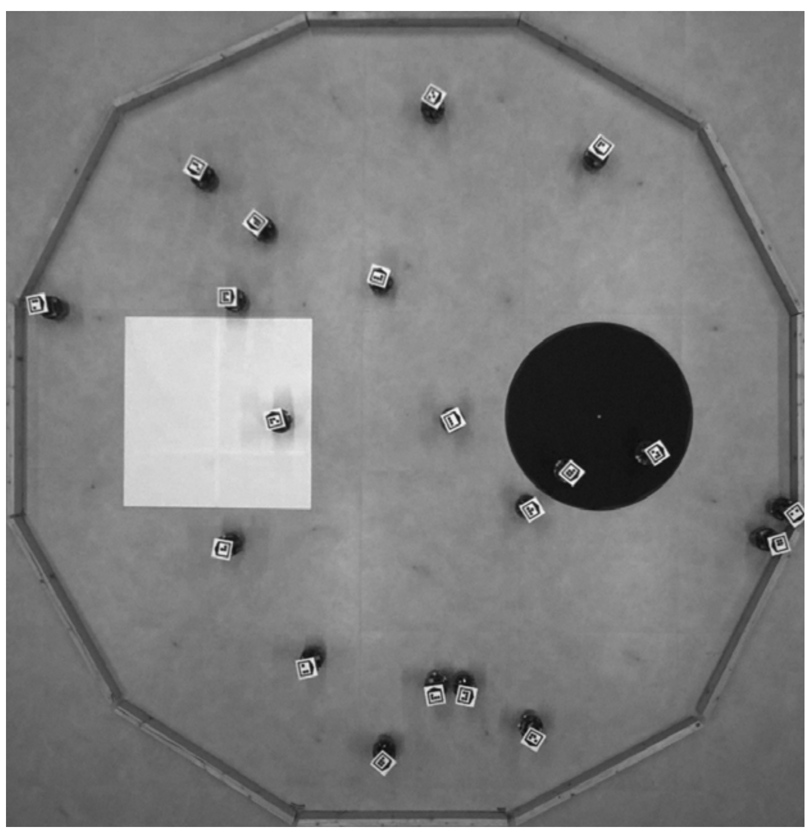




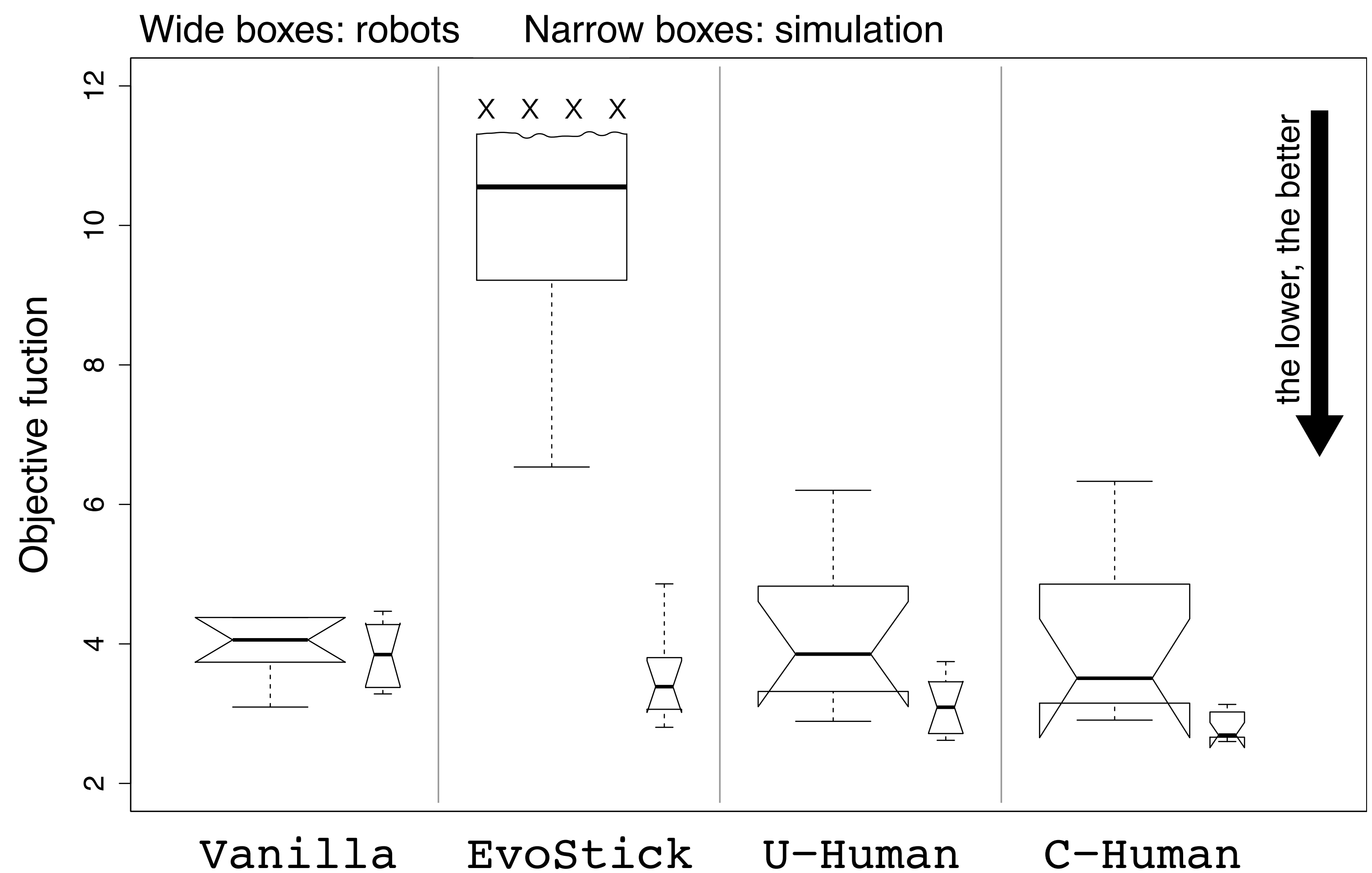
$F = \mathrm{E}[d(T)]$

CFA – coverage with forbidden areas: robot should cover the arena except the forbidden black areas

mission and results

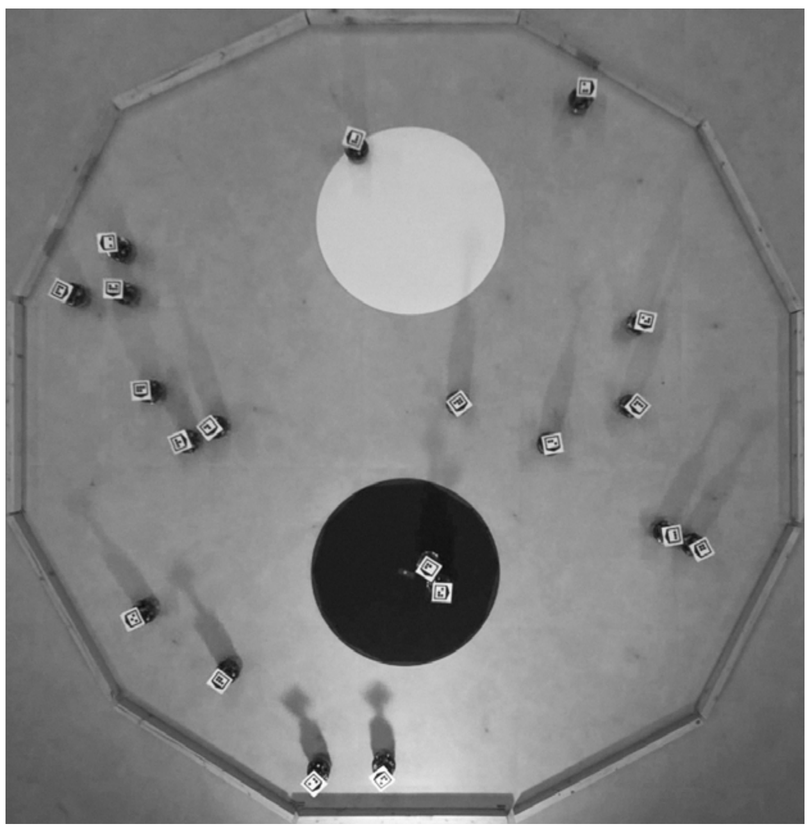


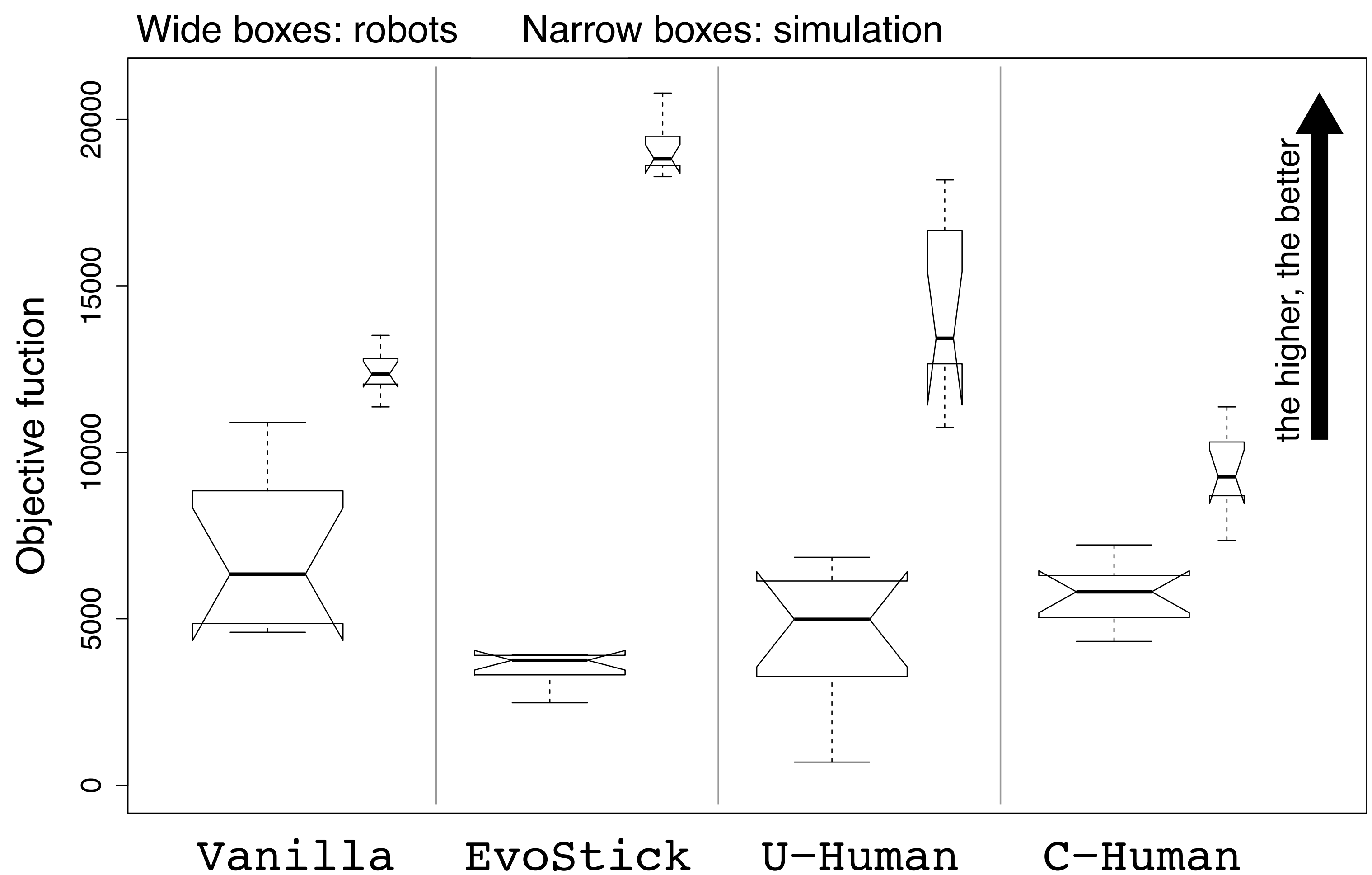
$F = \mathrm{E}[d_a(T)]/c_a + \mathrm{E}[d_p(T)]/c_p$



SPC – surface and perimeter coverage: robot should cover the surface of the white square and the perimeter of the black circle

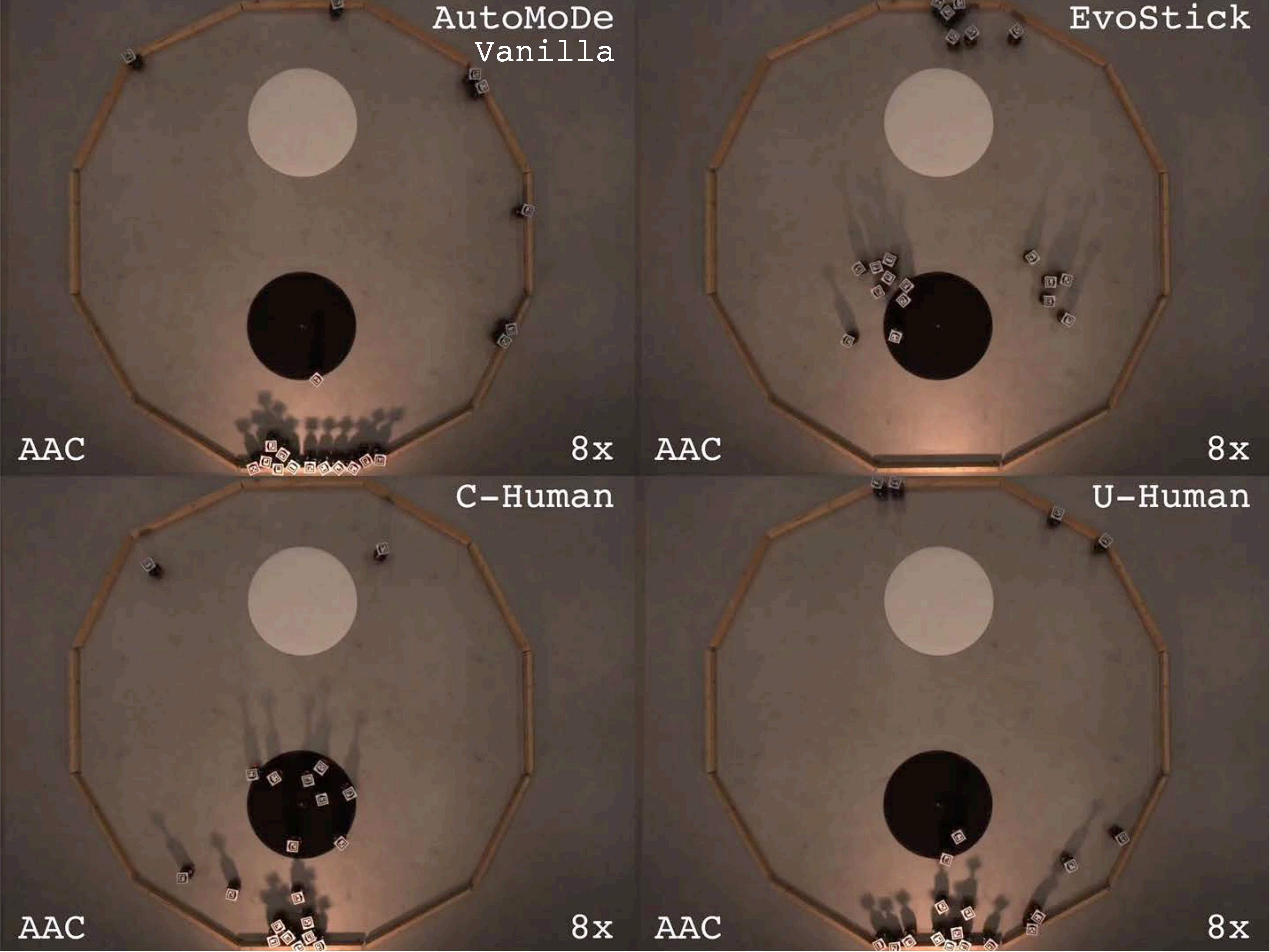
mission and results





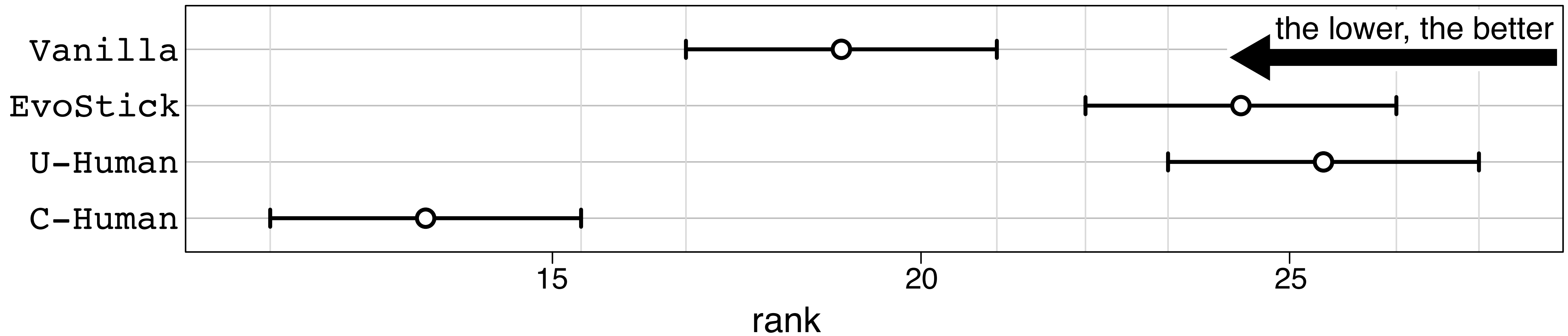
$$F = \sum_{t=1}^{T} N_b(t)$$

AAC – aggregation with ambient cues: robot should aggregate in the black region


AutoMoDe
Vanilla
AAC
8x
EvoStick
AAC
8x
C-Human
AAC
8x
U-Human
AAC
8x

aggregate results

analysis based on Friedman test

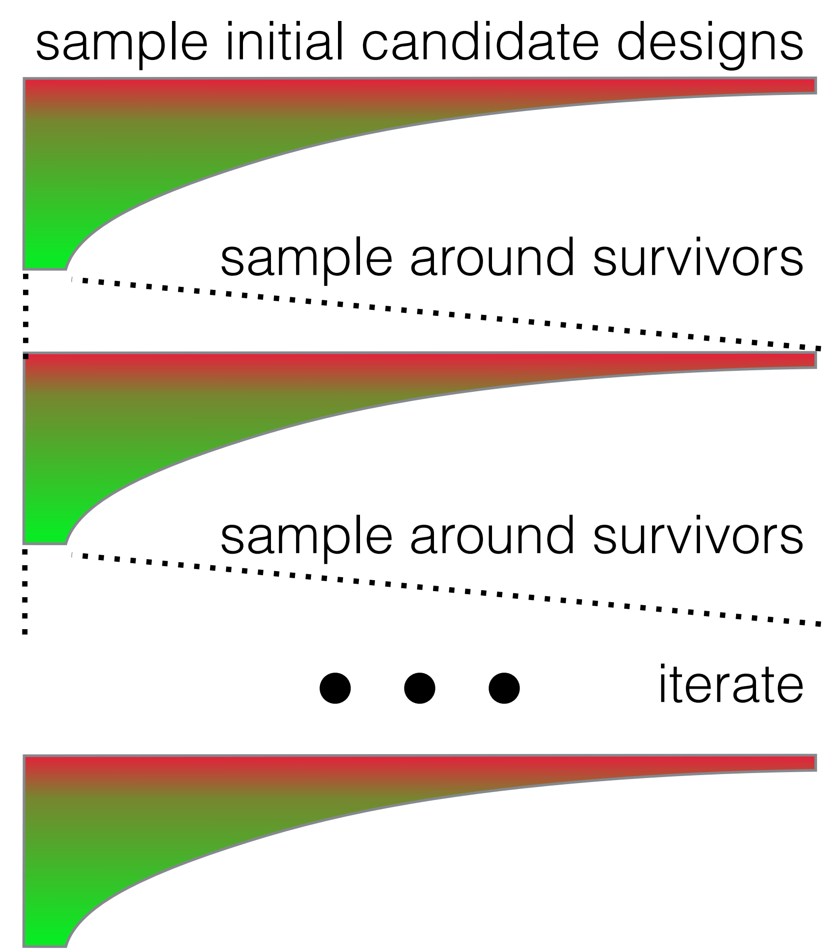


- **`vanilla`** confirms better that **`EvoStick`** also on new missions
- **`C-Human`** is the best: vanilla's modules are good but F-Race is unable to fully exploit their potential
- humans experience the reality gap as well as automatic methods
- injecting bias (by constraining the human to the given modules) improves the ability to cross the reality gap also for humans

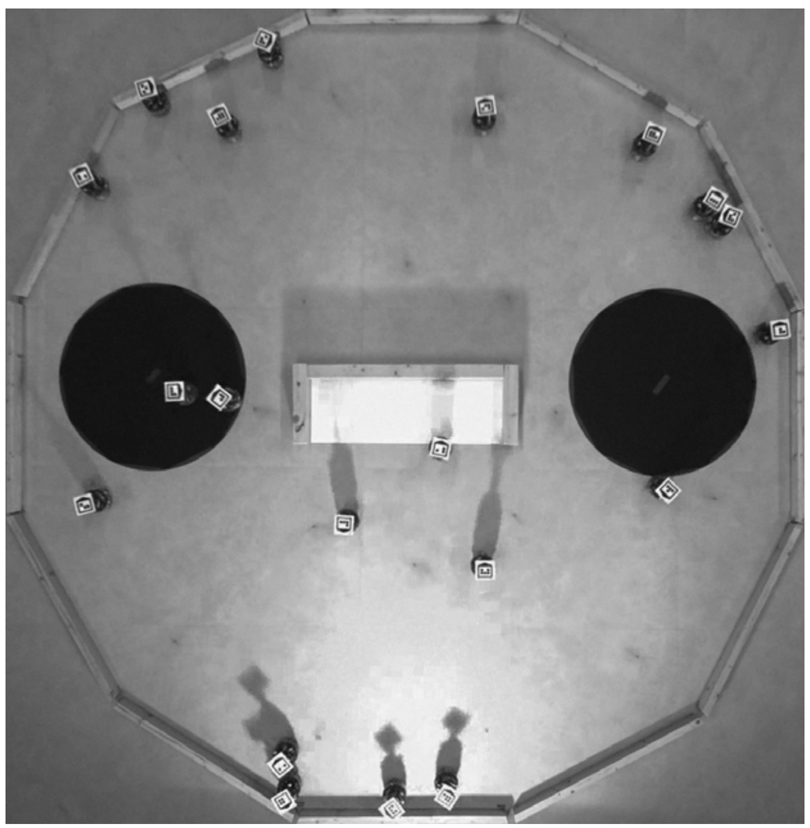
AutoMoDe–**`chocolate`**

same modules of **`vanilla`** (and **`C-Human`**), but
new optimization algorithm: iterated F-Race (Balaprakash *et al.*, 2002)

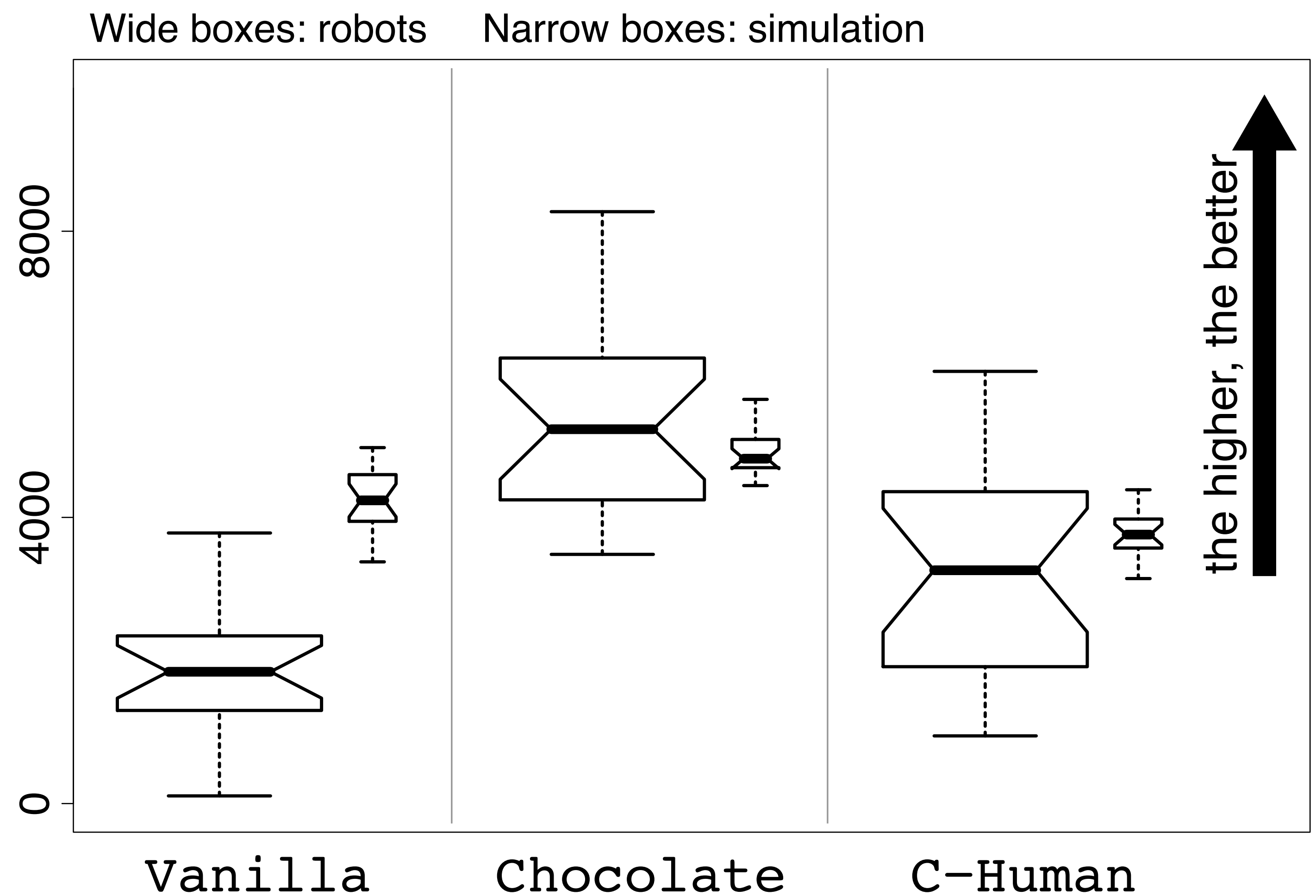
- sample candidate designs
- run F-Race
- sample around survivors
- run F-Race and iterate
- stop when budget depletes



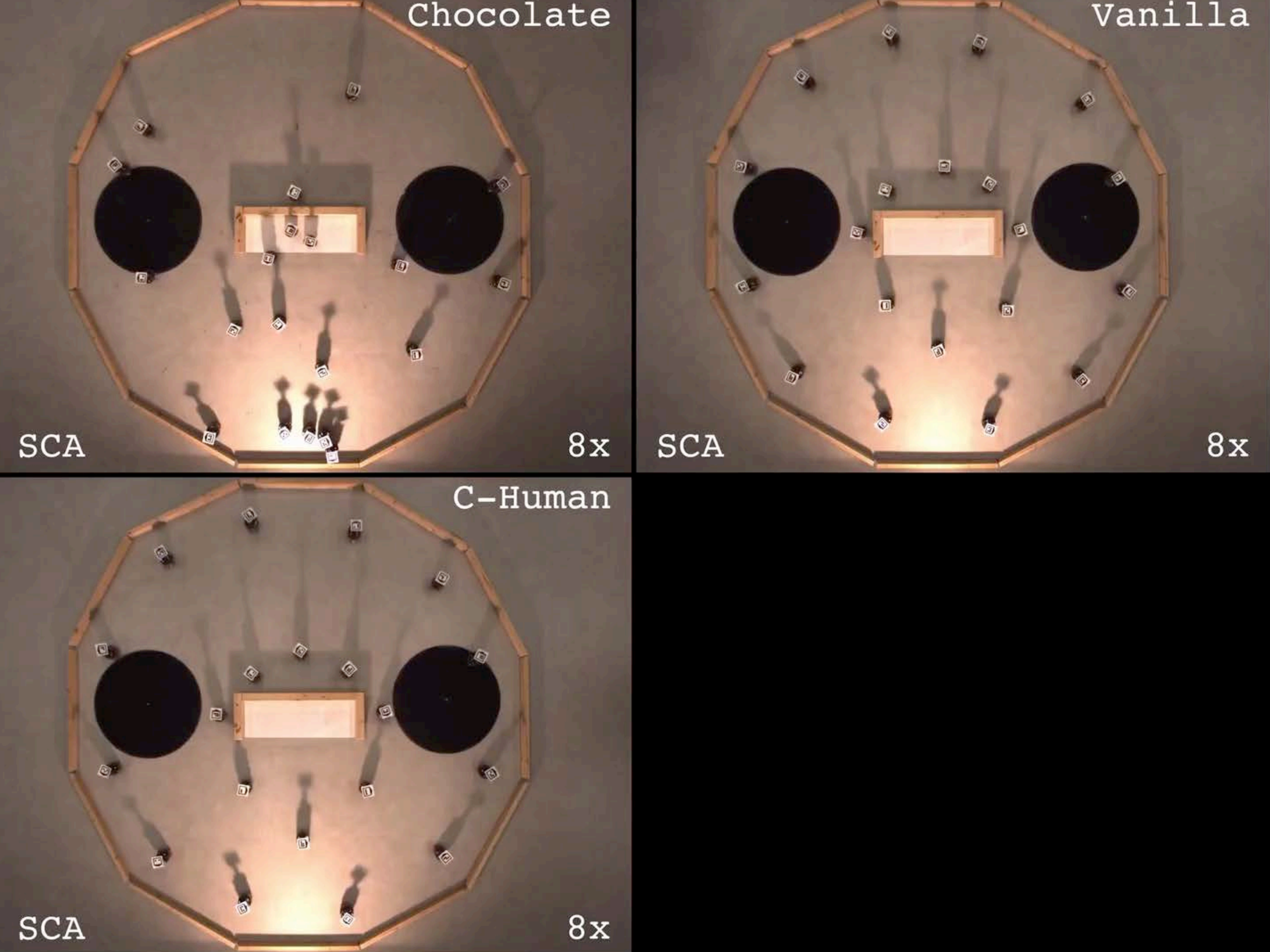
mission and results



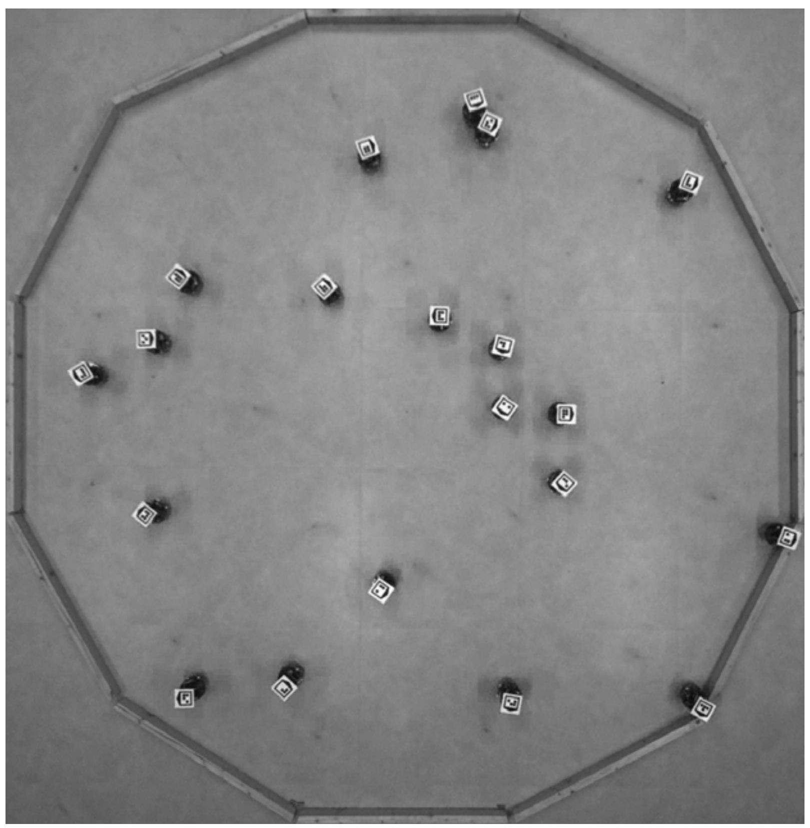
$F = \sum_{t=1}^{T} N(t)$



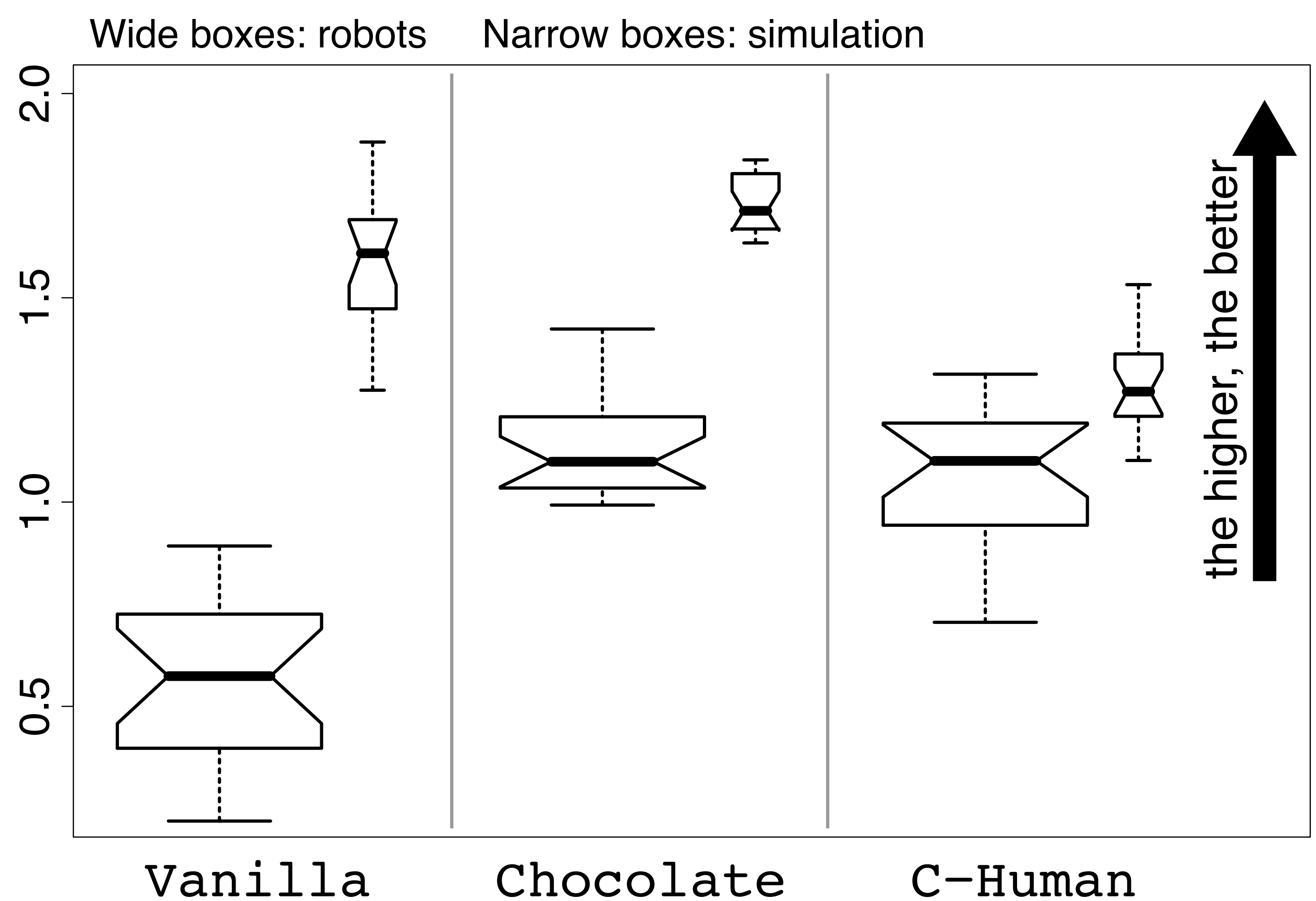
SCA – shelter with constrained access: robots should aggregate in the white shelter


Chocolate
SCA
8x
Vanilla
SCA
8x
C-Human
SCA
8x

mission and results

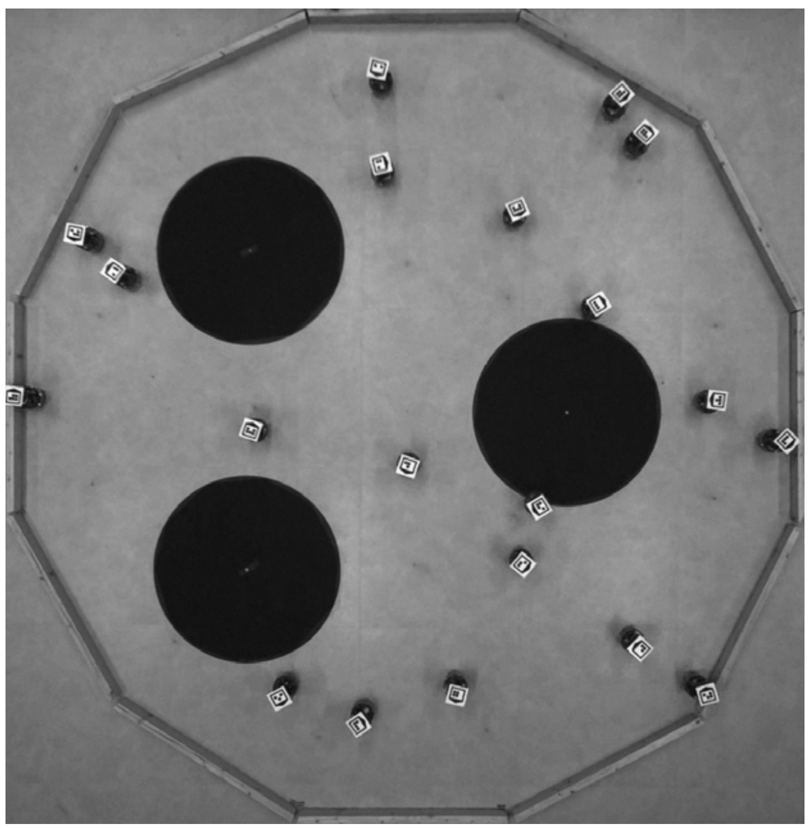


$F = A_{C(T)}$

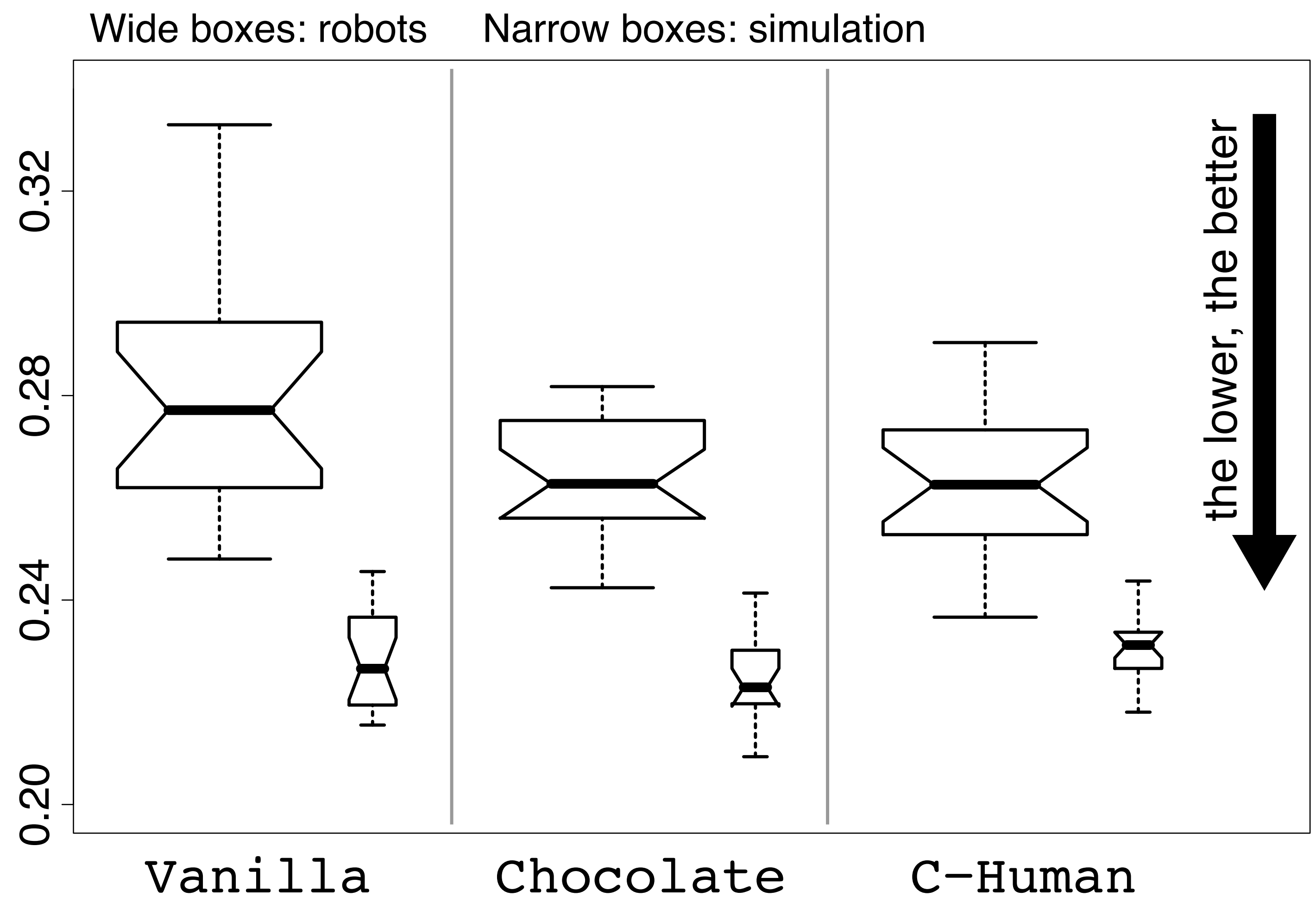


LCN – largest covering network: robots should cover the largest possible area while maintaining connection with one another

mission and results

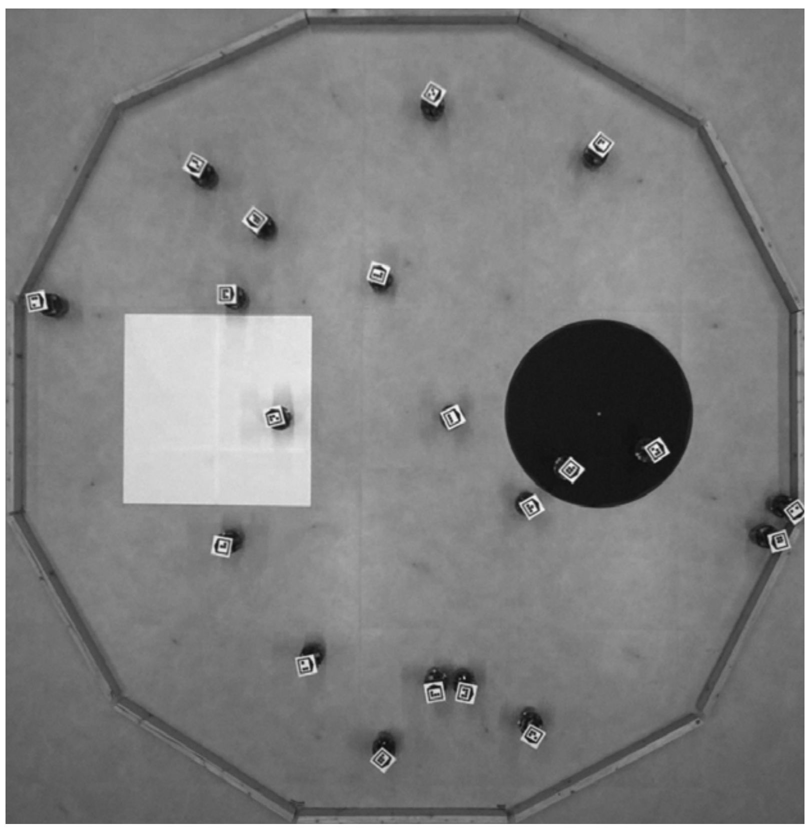


$F = \mathrm{E}[d(T)]$

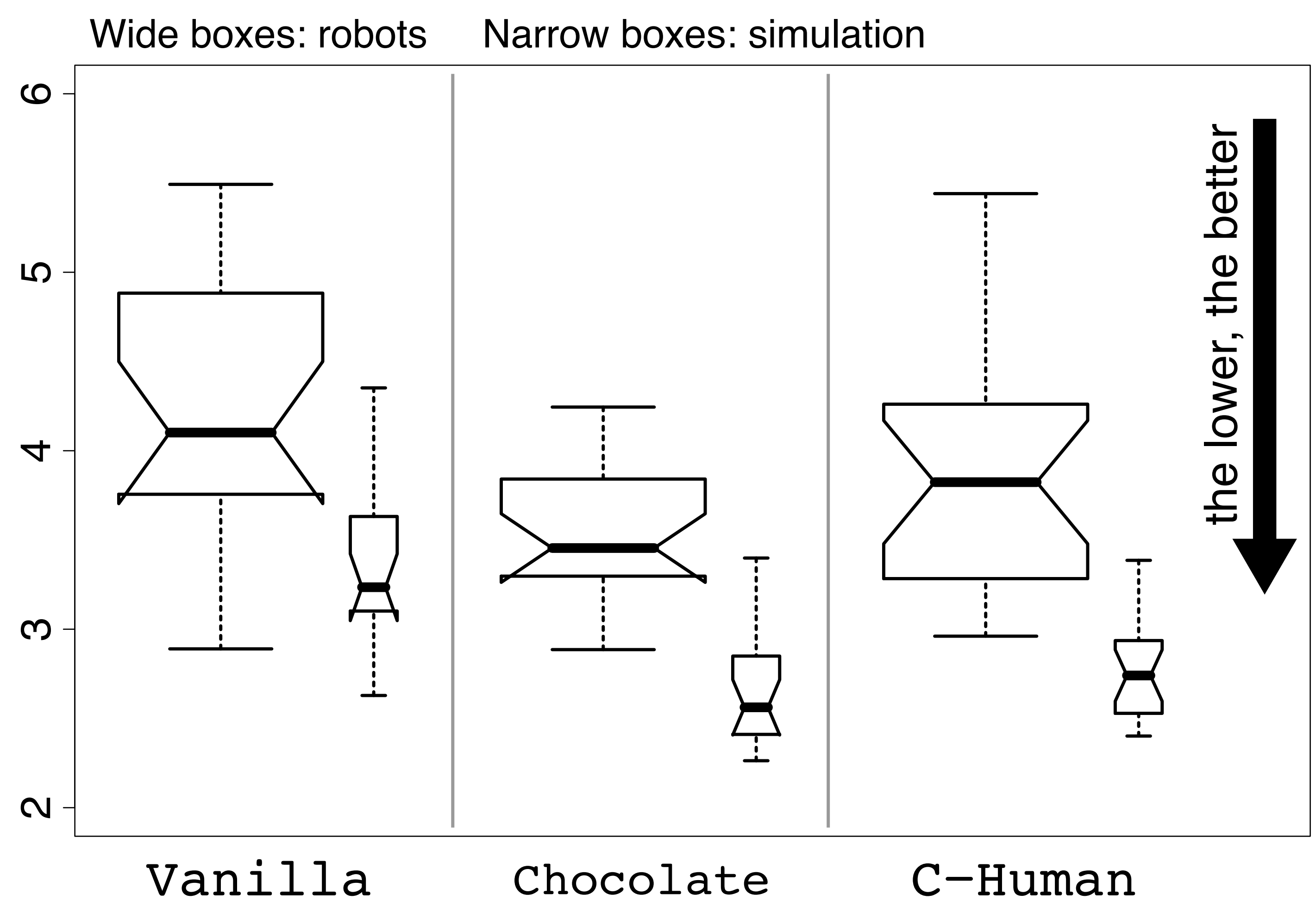


CFA – coverage with forbidden areas: robot should cover the arena except the forbidden black areas

mission and results

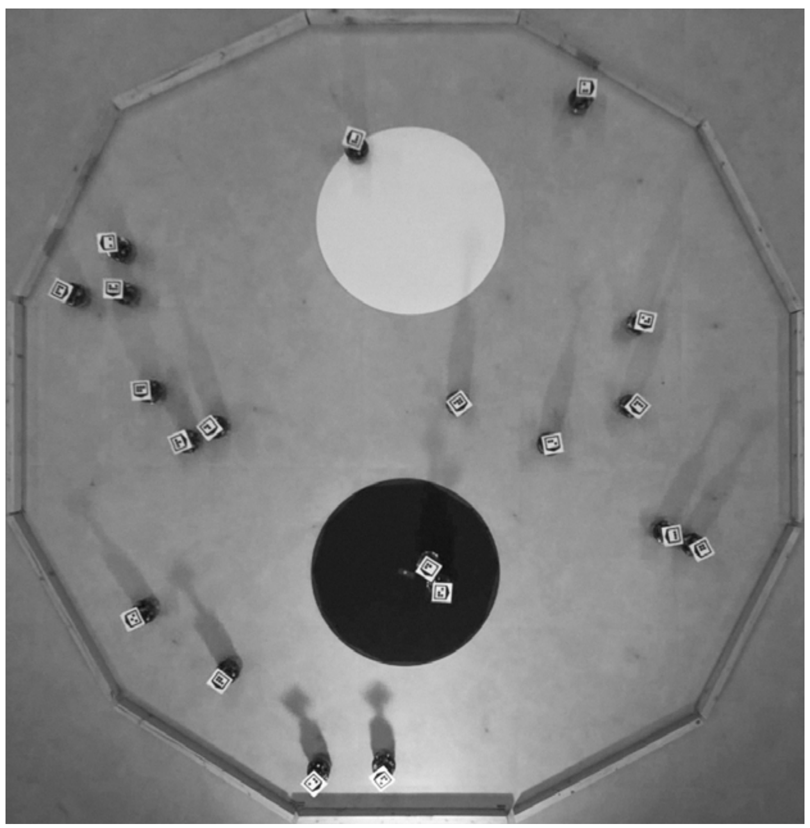


$F = E[d_a(T)]/c_a + E[d_p(T)]/c_p$

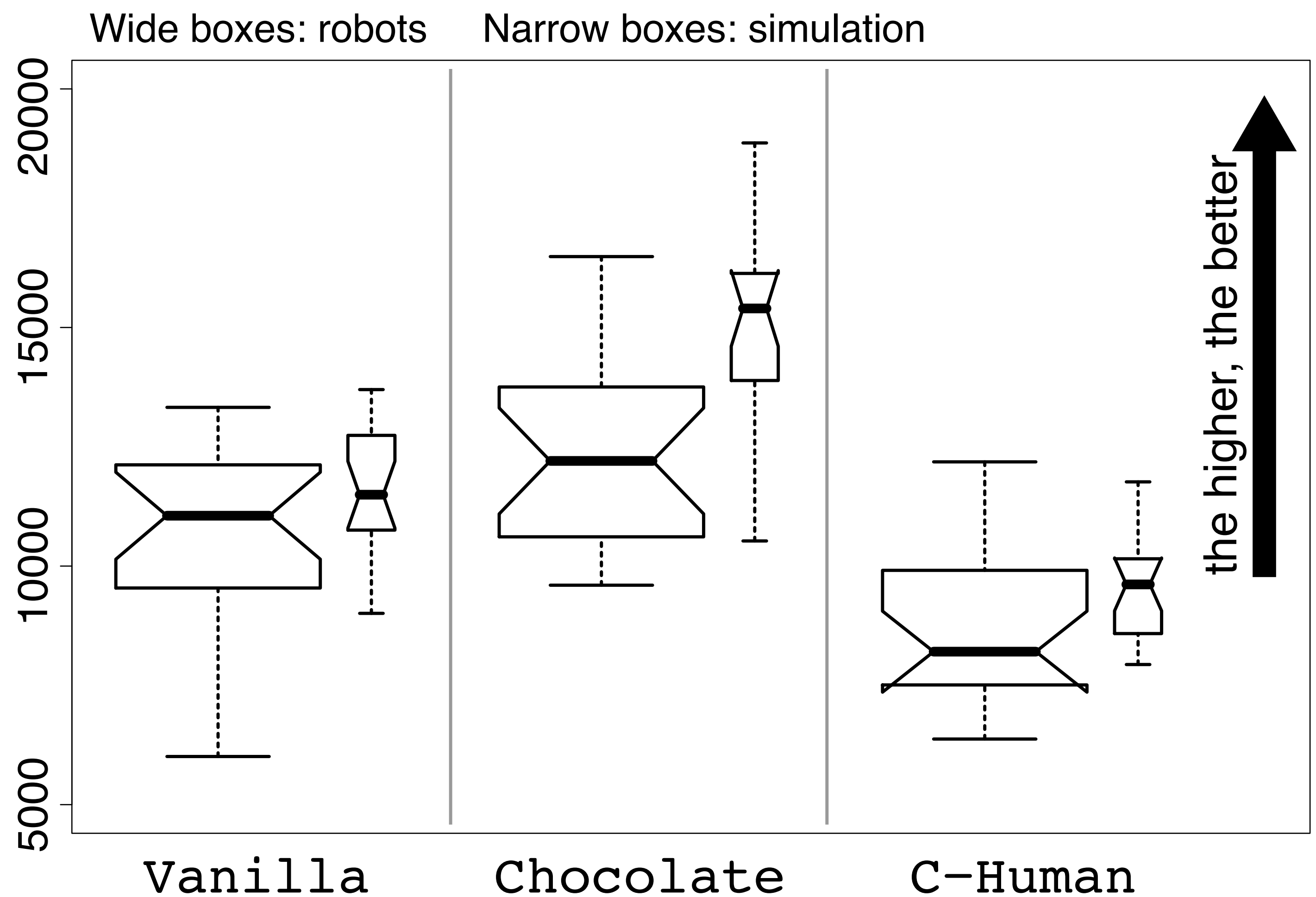


SPC – surface and perimeter coverage: robot should cover the surface of the white square and the perimeter of the black circle

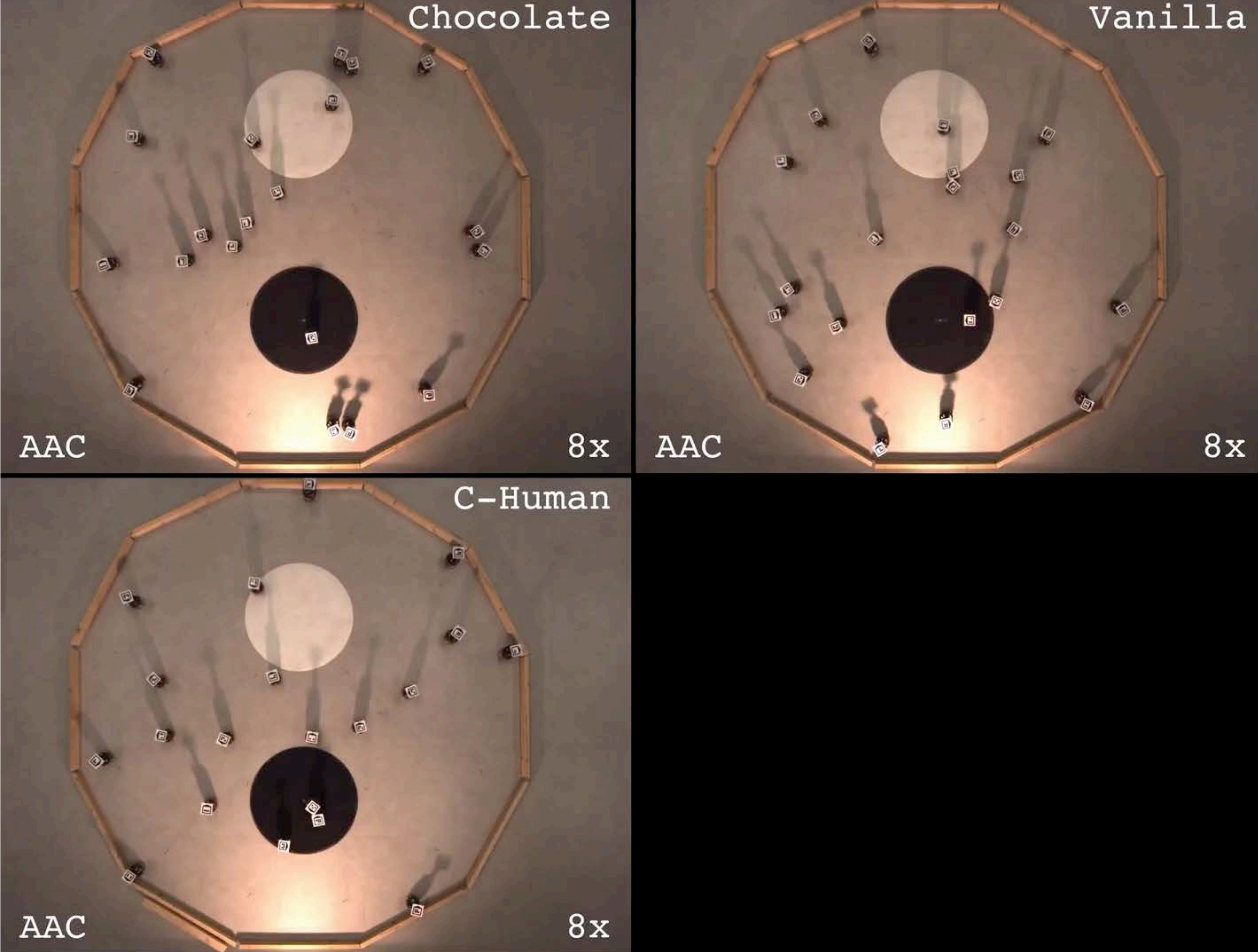
mission and results



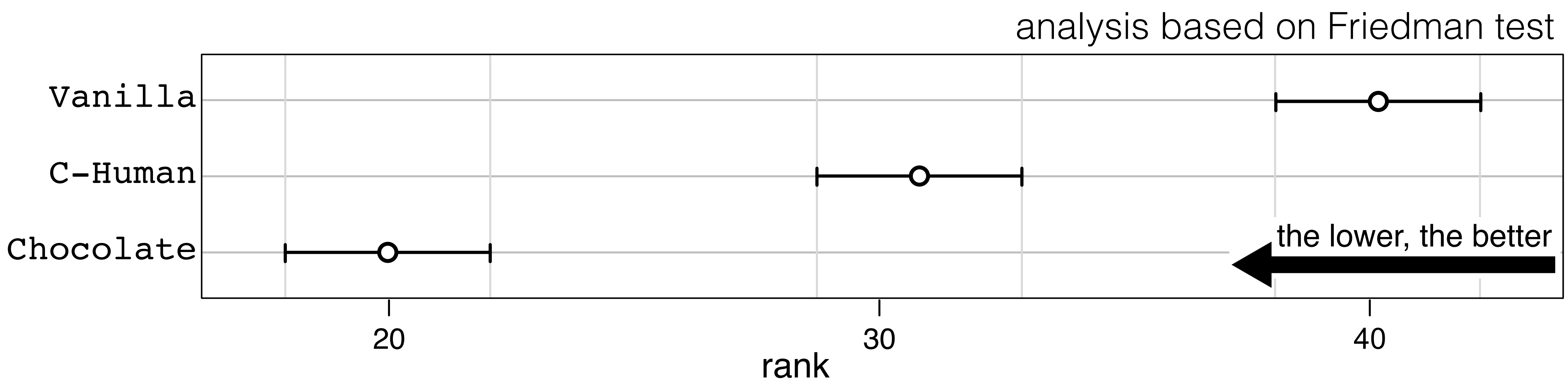
$F = \sum_{t=1}^{T} N_b(t)$



AAC – aggregation with ambient cues: robot should aggregate in the black region


Chocolate
AAC
8x
Vanilla
AAC
8x
C-Human
AAC
8x

aggregate results



- **`chocolate`** improves over **`vanilla`**
- **`chocolate`** performs better than **`C-Human`**

conclusions

- automatic design of robot swarms in the light of machine learning concepts
- AutoMoDe is a promising approach
- AutoMoDe-**`chocolate`** performed better that human designers
- some innovative elements in the empirical studies

beyond `vanilla` and `chocolate`...

`Gianduja` explicit communication

`Waffle` design of hardware and software

`Maple` behavior-tree architecture

`Mate` spatially-organizing behaviors

`TuttiFrutti` interaction via colors

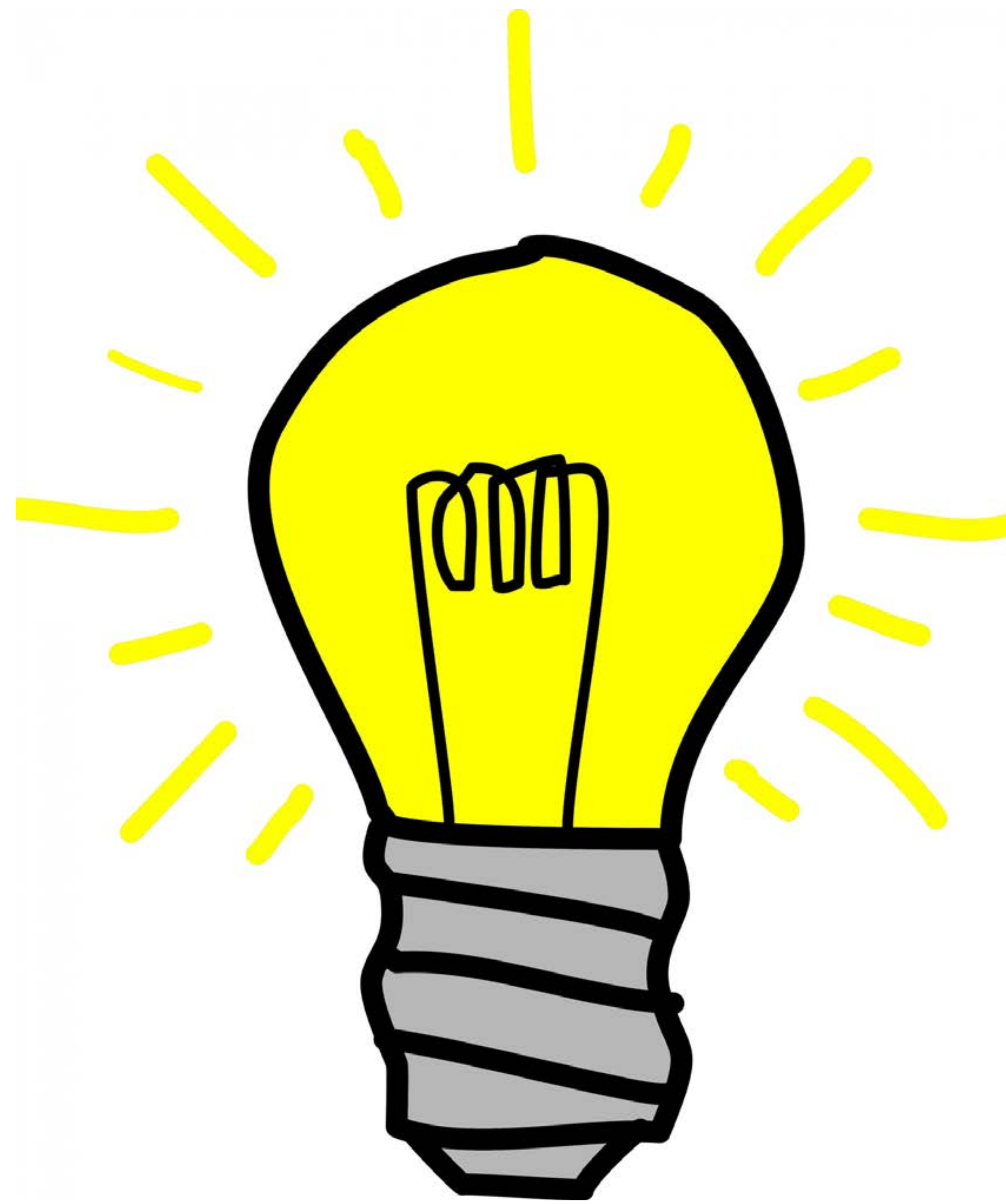
`Coconut` adaptive exploration

`IcePop` optimization via simulated annealing

other research directions

in the automatic design of robot swarms

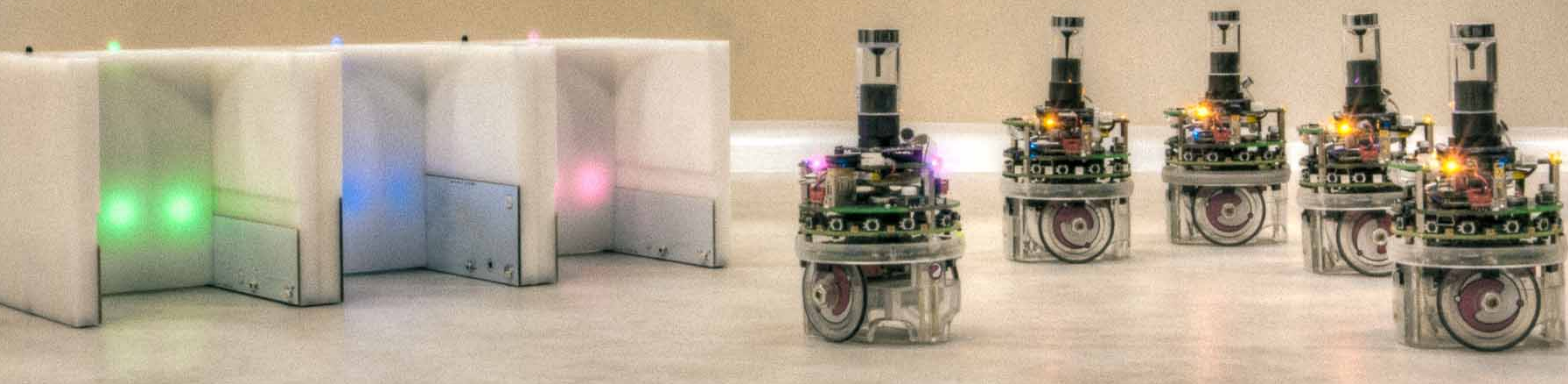
- understanding the reality gap
- formal specification of missions
- alternative optimization algorithms
- alternative software architectures
- scaling-up in complexity:
  - robot hardware/capabilities
  - low-level behaviors/conditions
  - missions



European Research Council
Established by the European Commission

DEMIURGE

Automatic Design of Robot Swarms



Université Libre de Bruxelles
IRIDIA
Institut de Recherches Interdisciplinaires
et de Développements en Intelligence Artificielle

European Commission

Plato's Demiurge in a drawing by the English pre-romantic poet and illustrator William Blake (1757-1827)

Demiurge: an intelligent system that designs robot swarms in an integrated and automatic way

starting from requirements expressed in an appropriate specification language, the Demiurge designs hardware and control software

the Demiurge does not create designs from scratch: it operates on preexisting software and hardware modules

contributors

Gianpiero Francesca
Manuele Brambilla
Arne Brutschy
Lorenzo Garattoni
Roman Miletitch
Gaëtan Podevijn
Andreagiovanni Reina
Touraj Soleymani
Mattia Salvaro
Carlo Pinciroli

Franco Mascia
Vito Trianni
Ken Hasselmann
Antoine Ligot
David Garzon Ramos
Miquel Kegeleirs
Jonas Kuckling
Muhammad Salman
Darko Bozhinoski
Federico Pagnozzi