Self-driven rewards for

Autonomous Robots: An information-theoretic Approach

Michèle Sebag

TAO

Joint work with Marc Schoenauer, Pierre Delarboulas

ICTAI 2010

Where are we going ?

The 2005 DARPA Challenge

AI Agenda: What remains to be done — Thrun 2005

- Reasoning — 10%
- Dialogue — 60%
- Perception — 90%

Where are we going ?

The 2005 DARPA Challenge

AI Agenda: What remains to be done — Thrun 2005

- Reasoning — 10%
- Dialogue — 60%
- Perception — 90%

This talk: About motivations/autonomy

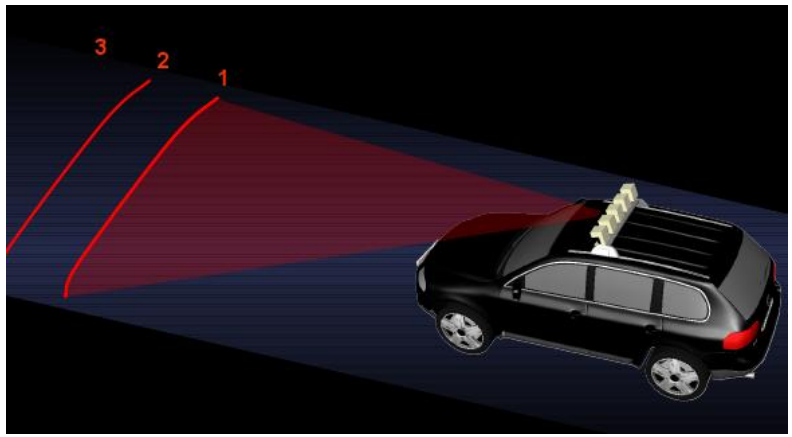
Overview

1. Why Robotics
2. Why Swarms
3. Challenges and Examples
4. The complete agent principles
5. An information-theoretic approach

WHY Robotics

Cognitive Systems — Lifelong learning

- Robotics is the ultimate work bench for AI



Challenges — for Machine Learning

- Non "independently identically distributed" data
- Right prediction $\not\rightarrow$ Best action
- Exploration vs Exploitation
- Active Learning and Robot Integrity

Challenges for Robotics

Battery

Motors

Software

The SYMBRION IP

2008-2013

http://symbrion.org/

Why Swarms ?

Emergence

- Simple agents — simple micro-motives for macro-behaviors
- No pacemakers — decentralized, distributed, randomized systems
- More is different

WHY

- inexpensive
- reliable
- ...Hayek's inheritage ?

The Rules of Swarms, 2

	Local information I_ℓ	$\longrightarrow$	estimates global quantities I
Intuition	Local information	$\longrightarrow$	individual behaviour $b(I_\ell)$
	Aggregate $b(I_\ell)$	$=$	Behaviour$[I]$

Examples

- Sounds & clusters of birds and frogs; Melhuish 99
- Bees & air-conditioning of the hive Auman 08

The Rules of Swarms, 2

	Local information l_ℓ	$\longrightarrow$	estimates global quantities l
Intuition	Local information	$\longrightarrow$	individual behaviour $b(l_\ell)$
	Aggregate $b(l_\ell)$	$=$	Behaviour$[l]$

Examples

- Sounds & clusters of birds and frogs; Melhuish 99
- Bees & air-conditioning of the hive Auman 08

From observing to designing emergence

Main Issues

- Communication
- Convergence
- Reality Gap
- Bootstrapping

Issue 1. Communication

Example: Potential fields

Reif & Wang 99

- Electric-like attraction/repulsion forces

$$f = -\frac{q_1 q_2}{d^2}$$

- Coordinate moves among (groups of) robots

CONS

- Oscillations near obstacles or in narrow passages
- Positioning needed

`http://www.inl.gov/adaptiverobotics/robotswarm`

Stigmergy is an alternative

Bonabeau et al. 00

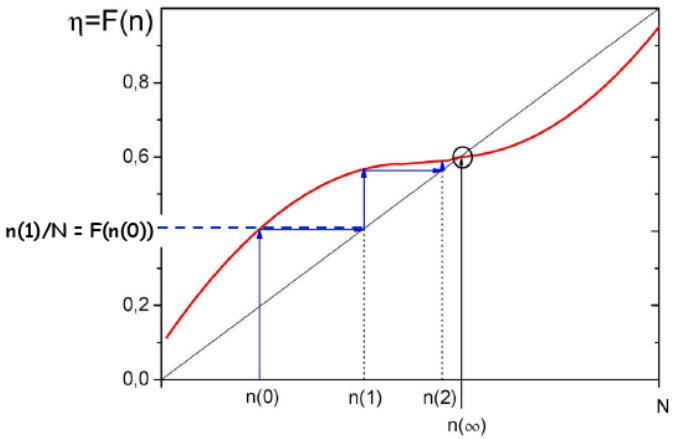
Implicit communication among robots through modification of environment.

Issue 2. Convergence

The dying seminar

Schelling, 1978; Nadal et al. 2009

- N scientists are asked to go to a seminar:
- ... scientist i will go if #attendees $> n(i)$

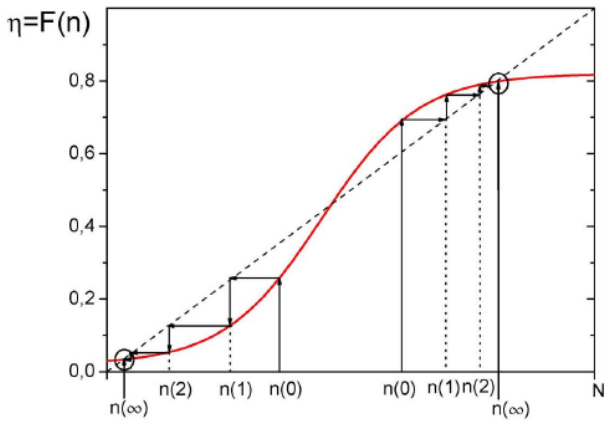


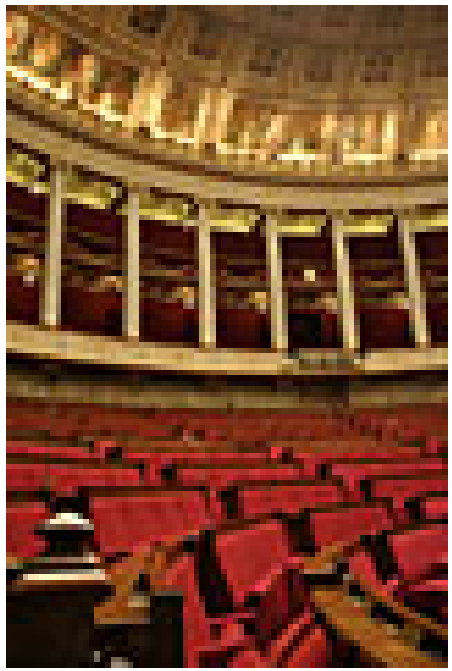
Major issues: 2. Convergence

The dying seminar

Schelling, 1978; Nadal et al. 2009

- N scientists are asked to go to a seminar:
- ... scientist i will go if #attendees $> n(i)$





Train in simulation, transfert on real robot

Simulator drawbacks

- Cheap and inaccurate, or takes for ever — ODE
- NB: most simulation time spent in simulating events which are not part of solution (hitting your leg) — Jacobi 94
- What reliability anyway ? — sensor noise model

The Reality Gap

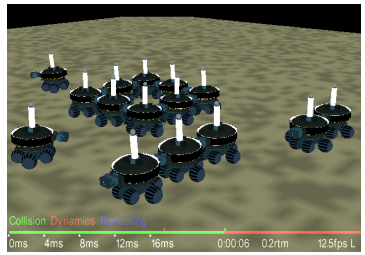
What works in simulation, does not on the robot

Closing the gap

- Ensemble of Simulators $\rightarrow$ Ensemble of Controllers $\rightarrow$ Controller Dimensionality Reduction — Kolter et al. 07
- Self-aware computational devices — Bongard et al. 06; Lipson et al. 09

Issue 3. Bootstrapping

Swarm-bot (2001-2005)
Swarmanoid (2005-2010)



- Phototaxis and hole avoidance
- Variety of sensors
- Multi-layer perceptron, 2 hidden neurons

Incremental evolution/task decomposition

- Acquire competence
- Address Reality Gap

Christensen & Dorigo

Fails

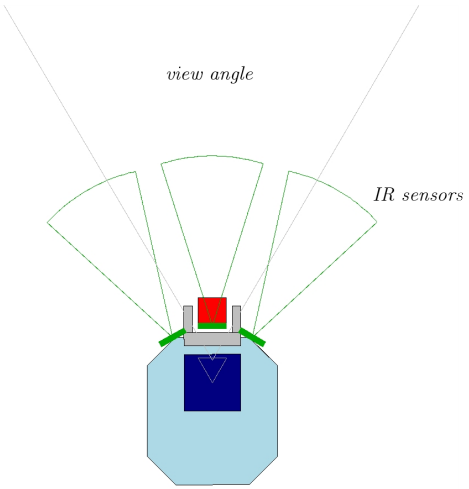
- Dynamic evolution was deceptive...

Adaptive Foraging in Swarm Robotic Systems

cleaning, harvesting, search and rescue, land-mine clearance, planetary exploration...

Simple agents
Finding food/resting

- Finding food delivers energy
- Searching costs energy
- bumping into other robots costs energy



Goal

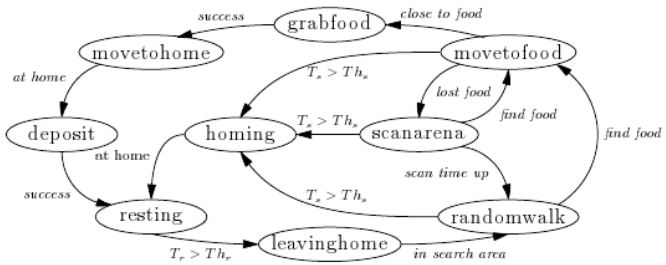
- Allocate time between search and rest

Alan Winfield & Wenguo Liu 08

`http://www.brl.uwe.ac.uk/projects/swarm/index.html`

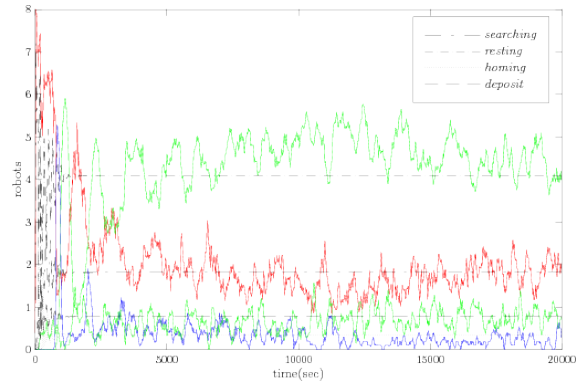
Adaptive Foraging in Swarm Robotic Systems, 2

Probabilistic Finite State Machine



Design

- Find transition probabilities
- Rest and Search thresholds
- Input:
  - internal cues (food retrieved)
  - environment cues (bumping into other robots)
  - social cues (success/failure of pals)



Overview

1. Why Robotics
2. Why Swarms
3. Challenges and Examples
4. The complete agent principles
5. An information-theoretic approach

The complete agent principles

Cognition

Transforming sensory data into motor commands
using internal representations

Therefore

Cognition lives in the sensori-motor loop
Clancey 1992; Clark 1996

Embodied AI

Brooks 86; Pfeifer & Bongard 2006

- Start with complete robotics systems of low complexity
- Make them work, understand why and increase complexity

Getting started

WHAT

- Learn about the world through acting

  intuition: motor or body babbling

HOW

- Reinforcement Learning — Sutton et al. 98; Peters et al. 2009
  - Define a reward function — per state
  - Learn an optimal policy: maximizing the cumulative reward
- Evolutionary Optimization — Nolfi & Floreano 2000
  - Define a fitness function — per behavior
  - Optimize a policy, getting maximal fitness.

Act before knowing it all

The challenge of designing a fitness

Nolfi & Floreano, 2000

1. Penalize the robot if it bumps into obstacles

The challenge of designing a fitness

Nolfi & Floreano, 2000

1. Penalize the robot if it bumps into obstacles

→ I don't move.

The challenge of designing a fitness

Nolfi & Floreano, 2000

1. Penalize the robot if it bumps into obstacles

$\rightarrow$ I don't move.

2. Move ! and go fast

The challenge of designing a fitness

Nolfi & Floreano, 2000

1. Penalize the robot if it bumps into obstacles

→ I don't move.

2. Move ! and go fast

→ I make circles !

The challenge of designing a fitness

Nolfi & Floreano, 2000

1. Penalize the robot if it bumps into obstacles

→ I don't move.

2. Move ! and go fast

→ I make circles !

3. Go fast and don't make circles !

The challenge of designing a fitness

Nolfi & Floreano, 2000

1. Penalize the robot if it bumps into obstacles

→ I don't move.

2. Move ! and go fast

→ I make circles !

3. Go fast and don't make circles !

→ I find a free way and go back and forth !

Shaping fitness: an interactive process

... It's hard to reward the robot based on what it does...

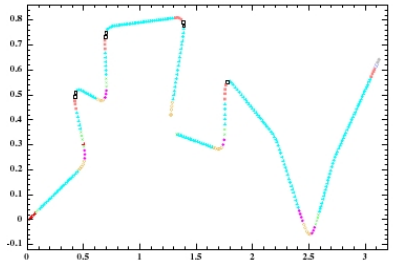
Becoming self-aware !

Schmidhuber 1990; Oudeyer & Kaplan 2007

Robot $\equiv$ a data stream

$$t \rightarrow x[t] = (sensor[t], motor[t])$$

$$\text{Trajectory} = \{x[t], t = 1 \ldots T\}$$



Robot trajectory

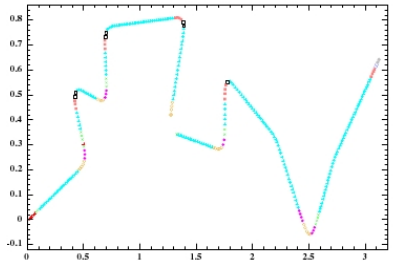
Becoming self-aware !

Schmidhuber 1990; Oudeyer & Kaplan 2007

Robot $\equiv$ a data stream

$$t \rightarrow x[t] = (sensor[t], motor[t])$$

$$\text{Trajectory} = \{x[t], t = 1 \dots T\}$$



Robot trajectory

Reward: Learn to anticipate

Forward model

$$(sensor[t], motor[t]) \rightarrow \widehat{sensor[t+1]}$$

$$Fitness = \sum_t ||sensor[t] - \widehat{sensor[t]}||^2$$

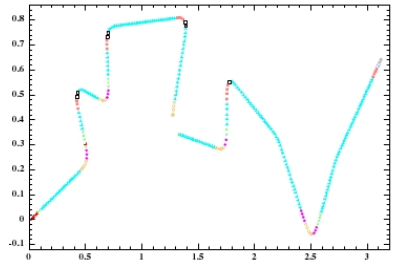
Becoming self-aware !

Schmidhuber 1990; Oudeyer & Kaplan 2007

Robot $\equiv$ a data stream

$$t \rightarrow x[t] = (sensor[t], motor[t])$$

$$\text{Trajectory} = \{x[t], t = 1 \dots T\}$$



Robot trajectory

Reward: Learn to anticipate

Forward model

$$(sensor[t], motor[t]) \rightarrow \widehat{sensor[t+1]}$$

$$Fitness = \sum_t ||sensor[t] - \widehat{sensor[t]}||^2$$

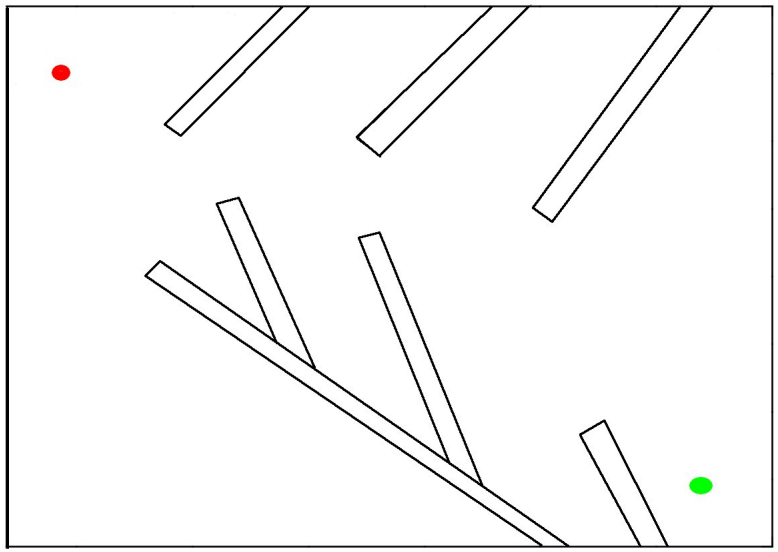
$\rightarrow$ I don't move, and predict $sensor[t+1] = sensor[t]$

See also Lungarella 2005; Oudeyer et al. 2008; Zahedi et al. 2010

Being different !

Intuition

- The search space of robot behaviors is huge
- Sample it in an automatic way
- Ask the designer to select the best behavior in the sample



Being different, cont'd

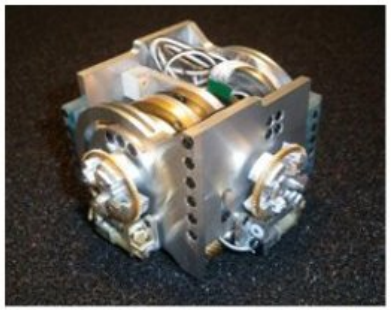
1. Fitness = diversity — Lehman & Stanley 2008
   - Maintain an archive;
   - Fitness = distance (of the ending point of the current trajectory) to the other ending points.

2. Multi-objective optimization — Mouret & Doncieux 2009
   - 1st objective is the (eye bird) distance to the goal
   - 2nd objective: diversity

Overview

1. Why Robotics
2. Why Swarms
3. Challenges and Examples
4. The complete agent principles
5. An information-theoretic approach

The vision



Requirements

1. No simulation
2. On-board training
   - Frugal (computation, memory)
   - No ground truth
3. Providing "interesting results"

"Human – robot communication"

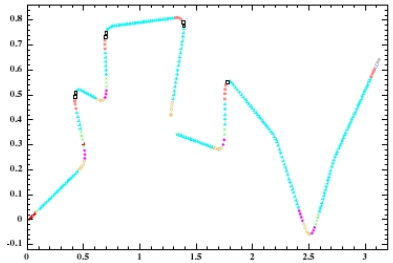
Goal: self-driven Robots : Defining instincts

Starting from (almost) nothing

Robot $\equiv$ a data stream

$$t \rightarrow x[t] = (sensor[t], motor[t])$$

$$\text{Trajectory} = \{x[t], t = 1 \ldots T\}$$



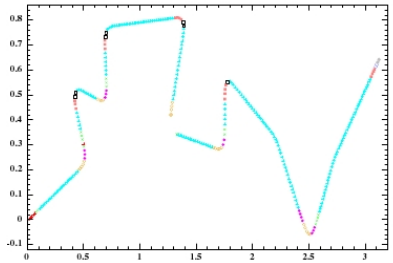
Robot trajectory

Starting from (almost) nothing

Robot $\equiv$ a data stream

$$t \rightarrow x[t] = (sensor[t], motor[t])$$

$$\text{Trajectory} = \{x[t], t = 1 \ldots T\}$$



Robot trajectory

Computing the quantity of information of the stream

Given $x_1, \ldots x_n$, visited with frequency $p_1 \ldots p_n$,

$$Entropy(trajectory) = -\sum_{i=1}^{n} p_i \log p_i$$

Conjecture

Controller quality $\propto$ Quantity of information of the stream

Building sensori-motor states

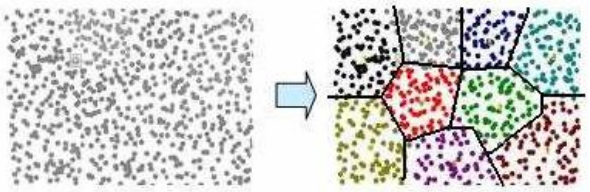
Avoiding trivial solutions...

If sensors and motors are continuous / high dimensional

- then all vectors $x[t]$ are different
- then $\forall i, p_i = 1/T$; $Entropy = \log T$

... requires generalization

From the sensori-motor stream to clusters



Clusters in sensori-motor space ($\mathbb{R}^2$)

sequence of points in $\mathbb{R}^d$

sensori-motor states

Trajectory $\rightarrow$

$x_1 x_2 x_3 x_1 \ldots$

Clustering

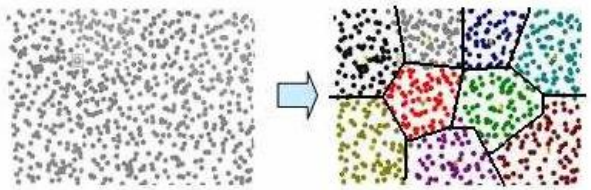
k-Means

1. Draw k points $x[t_i]$
2. Define a partition $\mathcal{C}$ in k subsets C_i *Voronoï cells*

$$C_i = \{x / d(x, x[t_i]) < d(x, x[t_j]), j \neq i\}$$

ϵ-Means

1. Init : $\mathcal{C} = \{\}$ *Initial site list*
2. For $t = 1$ to T *loop on trajectory*
   - If $d(x[t], \mathcal{C}) > \epsilon$, $\mathcal{C} \leftarrow \mathcal{C} \cup \{x[t]\}$



Curiosity Instinct

Search space

- Neural Net, 1 hidden layer.

Definition

- Controller F + environment $\longrightarrow$ Trajectory
- Apply Clustering on Trajectory
- For each C_i, compute its frequency p_i

$$\mathcal{F}(F) = -\sum_{i=1}^{n} p_i * \log(p_i)$$

Curiosity instinct: Maximizing Controller IQ

Properties

- Penalizes inaction: a single state $\longrightarrow$ entropy $= 0$
- Robust w.r.t. sensor noise (outliers count for very little)
- Computable online, on-board (use ϵ-clustering)
- Evolvable onboard

Limitations: does not work if

- Environment too poor
  (in desert, a single state $\longrightarrow$ entropy $= 0$)
- Environment too rich
  (if all states are distinct, *Fitness*(controller) $= \log T$)

both under and over-stimulation are counter-effective.

From curiosity to discovery

Intuition

- An individual learns sensori-motor states ($x[t_i]$ center of C_i)
- The SMSs can be transmitted to offspring
- giving the offspring an access to "history"
- The offspring can try to "make something different"

$$\text{fitness(offspring)} = \text{Entropy(Trajectory(ancestors} \bigcup \text{offspring))}$$

NB: does not require to keep the trajectory of all ancestors.
One only needs to store $\{C_i, n_i\}$

From curiosity to discovery

Cultural evolution transmits genome + "culture"

1. parent = (controller genome, $(C_1, n_1), \ldots (C_K, n_K)$)
2. Perturb parent controller $\longrightarrow$ offspring controller
3. Run the offspring controller and record $x[1], \ldots x[T]$
4. Run ϵ-clustering variant.

$$Fitness(offspring) = -\sum_{i=1}^{\ell} p_i \log p_i$$

ϵ-clustering variant

Algorithm

1. Init : $\mathcal{C} = \{(C_1, n_1), \ldots (C_K, n_K))\}$ *Initial site list*
2. For $t = 1$ to T *loop on trajectory*
   - If $d(x[t], \mathcal{C}) > \epsilon$, $\mathcal{C} \leftarrow \mathcal{C} \cup \{x[t]\}$
3. Define $p_i = n_i / \sum_j n_j$

$$Fitness(offspring) = -\sum_{i=1}^{\ell} p_i \log p_i$$

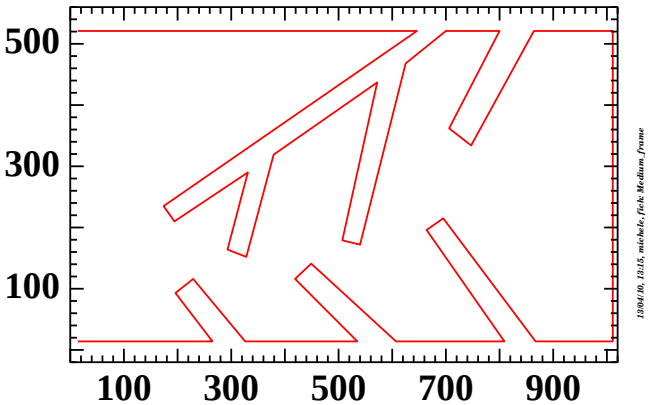
Validation

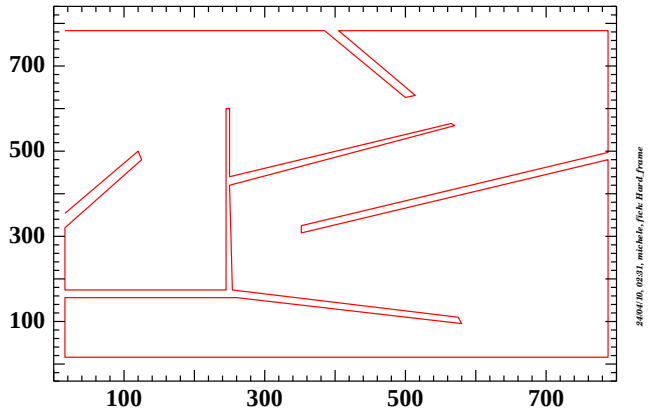
Experimental setting

Robot = Cortex M3, 8 infra-red sensors, 2 motors.

Controller space = ML Perceptron, 10 hidden neurons.

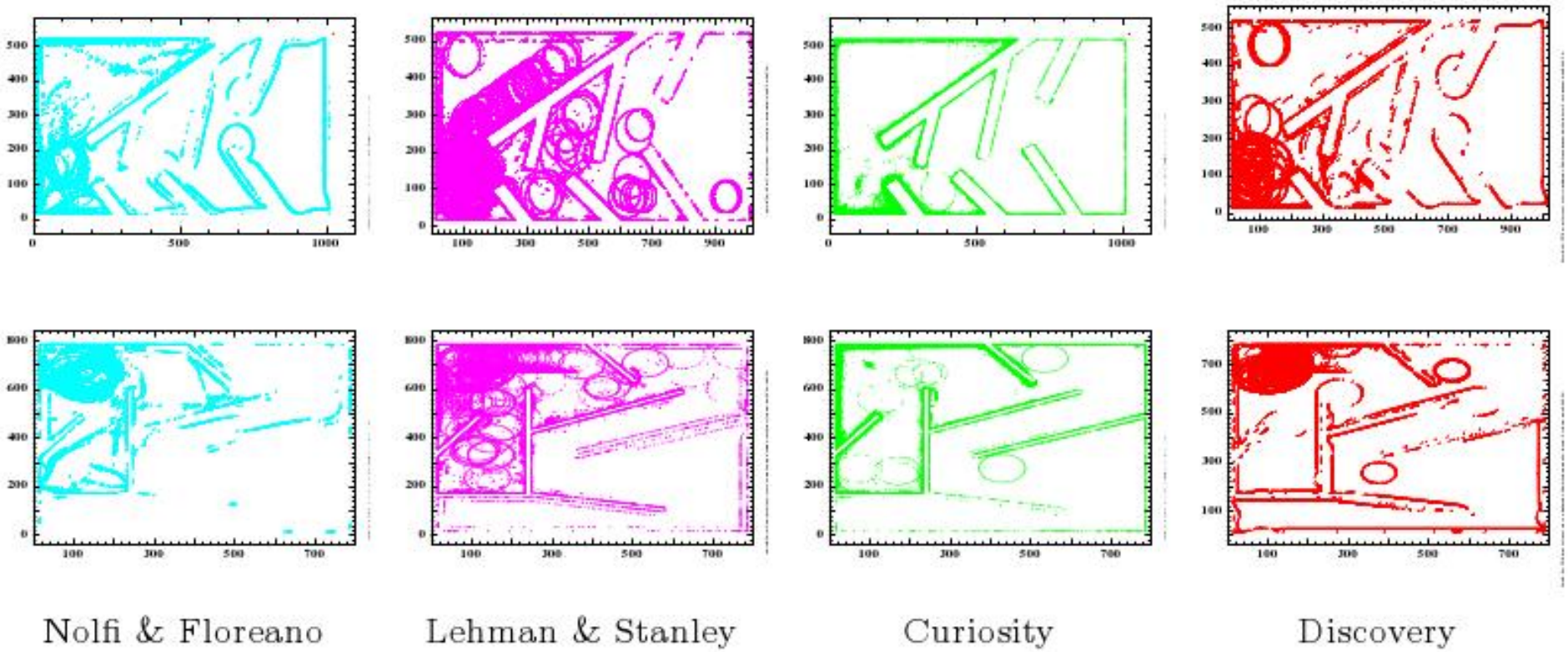
Medium and Hard Arenas





Validation, 2

Plot points in hard arena visited 10 times or more by the 100 best individuals.



Nolfi & Floreano | Lehman & Stanley | Curiosity | Discovery

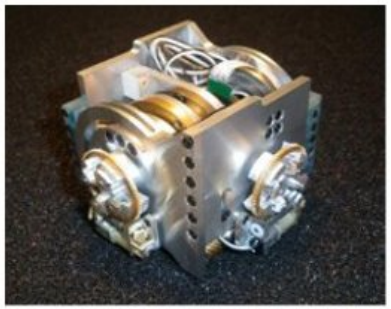
PPSN 2010

Discussion and Future work

The approach fits the requirements: computable on-board for swarm robot architecture; no need of prior knowledge/ground truth.
The robot only aims at gathering information, as opposed to, building a world model. (unsupervised as opposed to supervised learning).
The robot is rewarded for what it gets (sensory information) not what it does (e.g. going fast and not circling).

Caveat: needs a stimulating environment.

Vision and Perspectives



Requirements

- X No simulation
- X On-board training
  - Frugal (computation, memory)
  - No ground truth
- ? Providing "interesting results"

Human Robot Interaction

- Sensori-motor states can be interpreted, enabling the designer to take control, e.g. for emergency situations.
- Display trajectories: The designer can emit preferences

  Interactive optimization

Preference learning: associate "values" to sensori-motor states