Tutorial on Evolutionary Robotics

N. Bredeche[1,2], S. Doncieux[1,2], J.-B. Mouret[1−5]

[1] Sorbonne Universités, UPMC Univ Paris 06, UMR 7222, ISIR, F-75005, Paris, France
[2] CNRS, UMR 7222, ISIR, F-75005, Paris, France
[3] Inria, Villers-lès-Nancy, F-54600, France
[4] CNRS, Loria, UMR nº 7503, Vandœuvre-lès-Nancy, F-54500, France
[5] Université de Lorraine, Loria, UMR nº 7503, Vandœuvre-lès-Nancy, F-54500, France

bredeche@isir.upmc.fr, doncieux@isir.upmc.fr, jean-baptiste.mouret@inria.fr

http://wwww.sigevo.org/gecco-2015/

GECCO'15 Companion, July 11–15, 2015, Madrid, Spain.
ACM 978-1-4503-3488-4/15/07.
http://dx.doi.org/10.1145/2739482.2756583

Instructors

Nicolas Bredeche

- Professor at Pierre & Marie Curie University (UPMC), Paris, France
- Member of the AMAC team of the Institute of Intelligent Systems and Robotics (ISIR)
- Previously assistant professor at Univ. Paris XI, INRIA TAO team, member of the Symbion EU project

Stéphane Doncieux

- Professor at Pierre & Marie Curie University (UPMC), Paris, France
- Leader of the AMAC team of the Institute of Intelligent Systems and Robotics (ISIR)
- Coordinator of the EU project 'DREAM'

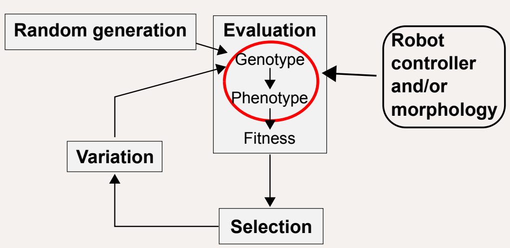
Jean-Baptiste Mouret

- Research Scientist at Inria - Nancy Grand-Est, France
- Previously assistant professor at UPMC
- PI of the ERC project 'ResiBots'

Definition

> "ER aims to apply evolutionary computation techniques to evolve the overall design or controllers, or both, for real and simulated autonomous robots"
>
> Vargas et al. 2014

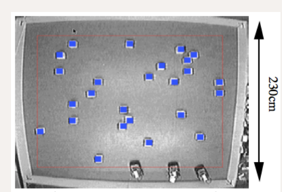
Principle

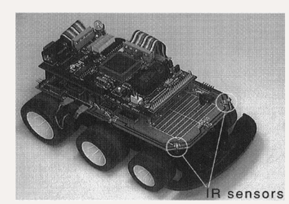


Motivation for roboticists

- Building robots with *embodied intelligence* [Pfeifer 2007]

Embodied Intelligence



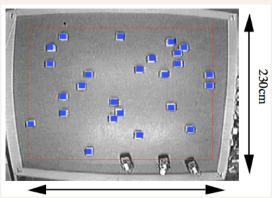


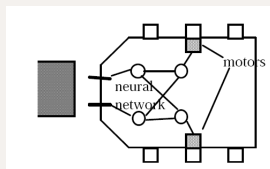
Pfeifer, R. and Bongard, J. (2007)
How the body shapes the way we think: a new view of intelligence,
MIT Press, Cambridge, MA.

Motivation for roboticists

- Building robots with *embodied intelligence* [Pfeifer 2007]

Embodied Intelligence





Pfeifer, R. and Bongard, J. (2007)
How the body shapes the way we think: a new view of intelligence,
MIT Press, Cambridge, MA.

Motivation for biologists

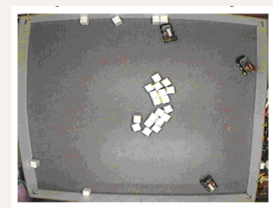
> "So far, we have been able to study only one evolving system and we cannot wait for interstellar flight to provide us with a second. If we want to discover generalizations about evolving systems, we have to look at artificial ones."
>
> John Maynard Smith, 1992

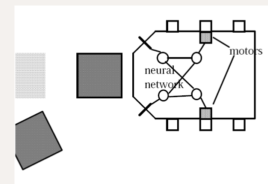
→ **computational modeling approach** based on agent based simulations including variation and selection mechanisms.

Motivation for roboticists

- Building robots with *embodied intelligence* [Pfeifer 2007]

Embodied Intelligence





Pfeifer, R. and Bongard, J. (2007)
How the body shapes the way we think: a new view of intelligence,
MIT Press, Cambridge, MA.

Main features of Evolutionary Robotics

Focus	control	and	morphology
Selective pressure	objective-driven	or	environment-driven
Implementation	simulation	or	real world
Space	centralized	or	distributed
Time	off-line	or	on-line

Main features of Evolutionary Robotics

Focus	control	and	morphology
Selective pressure	objective-driven	or	environment-driven
Implementation	simulation	or	real world
Space	centralized	or	distributed
Time	off-line	or	on-line

Focus of the tutorial

Evolutionary design of robot controller.

Main features of Evolutionary Robotics

Focus	control	and	morphology
Selective pressure	**objective-driven**	or	environment-driven
Implementation	simulation	or	real world
Space	centralized	or	distributed
Time	off-line	or	on-line

Outline of the tutorial

Part I. Selective pressures, S. Doncieux
What you should know about evaluation and selection to make an ER experiment successful.

Main features of Evolutionary Robotics

Focus	control	and	morphology
Selective pressure	objective-driven	or	environment-driven
Implementation	simulation	or	real world
Space	centralized	or	distributed
Time	off-line	or	on-line

Outline of the tutorial

Part II. Evolution for physical robots: the reality gap, J.-B. Mouret
How to make it work on real robots?

Main features of Evolutionary Robotics

Focus	control	and	morphology
Selective pressure	objective-driven	or	**environment-driven**
Implementation	simulation	or	real world
Space	centralized	or	**distributed**
Time	off-line	or	**on-line**

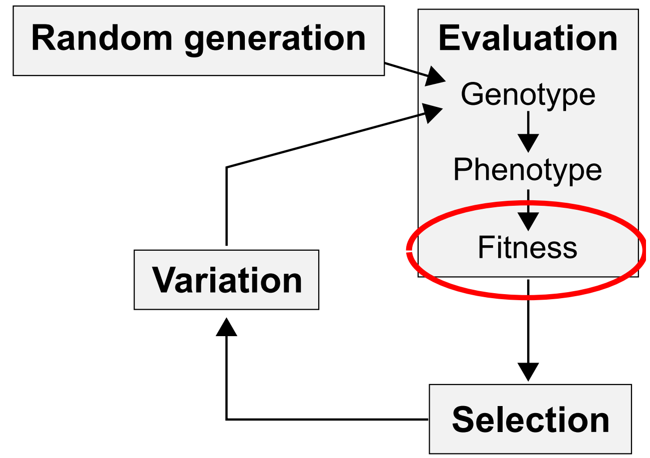
Outline of the tutorial

Part III. Embodied evolution and collective robotics systems, N. Bredeche
Evolution without a fitness for the design of distributed robotics systems and for modeling evolution of group dynamics.

Selective pressures in Evolutionary Robotics

ISIR, UPMC-CNRS
stephane.doncieux@isir.upmc.fr

GECCO 2015 Tutorial



Outline of this part of the tutorial

Introductory example
How to make a robot avoid obstacles ? Find the way out in a maze ?

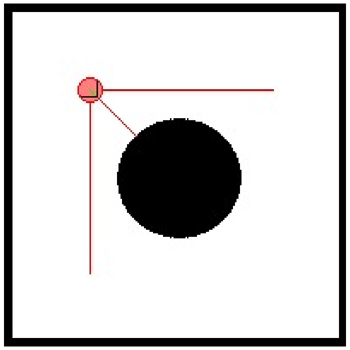
Definitions
Evaluation, different flavors of selective pressures

How to modify selective pressures ?
Overview of the main approaches to modify the selective pressures

Challenge : premature convergence
Process helpers to avoid premature convergence.

Introductory example

Example 1 : obstacle avoidance

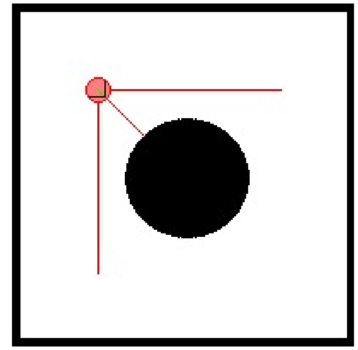


Setup

- NSGA-II, pop size=100, nb gen=1000
- Controller : feed-forward neural network
- Fitness : $\frac{1}{nb_{coll}+1}$

Source code on `http://www.isir.fr/evorob_db`

Example 1 : obstacle avoidance

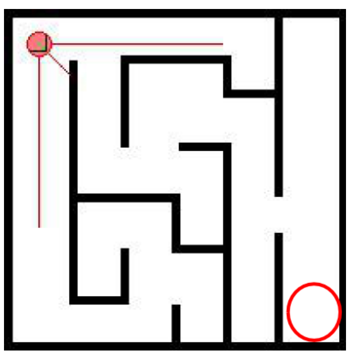


Setup

- NSGA-II, pop size=100, nb gen=1000
- Controller : feed-forward neural network
- Fitness : $\frac{1}{nb_{coll}+1} * \mathbf{avg_{speed}}$

Source code on `http://www.isir.fr/evorob_db`

Example 2 : maze navigation

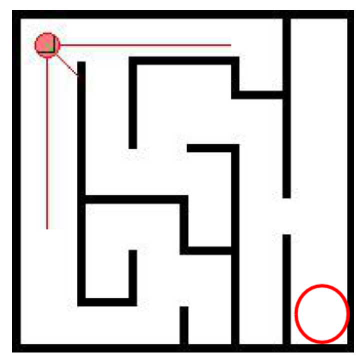


Setup

- NSGA-II, pop size=400, nb gen=8000
- Controller : neural network with an evolved topology
- Fitness : 1 if reached exit, 0 otherwise

Source code on `http://www.isir.fr/evorob_db`

Example 2 : maze navigation



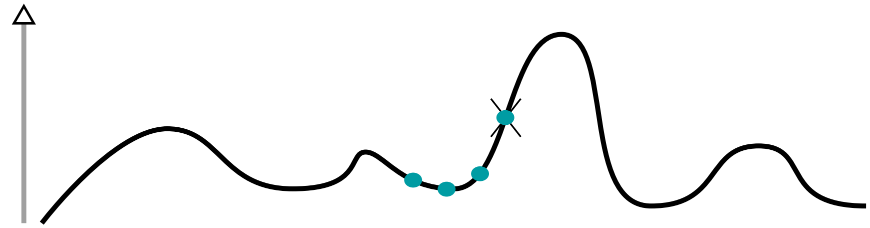
Setup

- NSGA-II, pop size=400, nb gen=8000
- Controller : neural network with an evolved topology
- Fitness : 1 if reached exit, 0 otherwise **+ another objective...**

Source code on `http://www.isir.fr/evorob_db`

Definitions

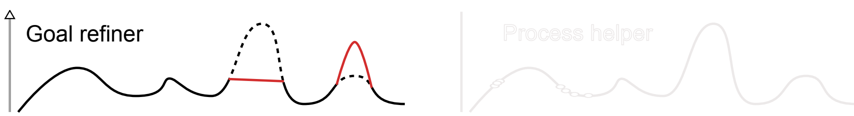
Selective pressures



The two facets of fitness functions

1. defines the goal
2. guides the search

Why modifying selective pressures ?



Goal refiners

A goal refiner aims at changing the optimum(s) of the fitness function by adding new requirements.

Process helpers

A process helper intends to increase the efficiency of the search process without changing the optimum(s) of the fitness function.

Task specific

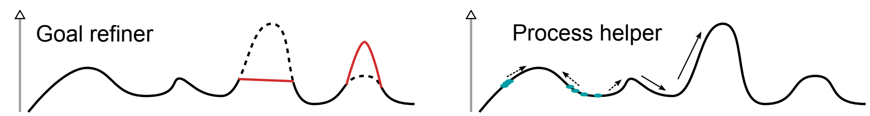
Task specific goal refiners/process helpers incorporate knowledge on how to solve the task.

Task agnostic

Task agnostic goal refiners/process helpers do not exploit knowledge about how to solve the task.

Doncieux, S. and Mouret J.-B. (2014).
Beyond Black-Box Optimization : a Review of Selective Pressures for Evolutionary Robotics.
Evol. Intel. DOI : 10.1007/s12065-014-0110-x, Springer Berlin Heidelberg, publisher.

Why modifying selective pressures ?



Goal refiners

A goal refiner aims at changing the optimum(s) of the fitness function by adding new requirements.

Process helpers

A process helper intends to increase the efficiency of the search process without changing the optimum(s) of the fitness function.

Task specific

Task specific goal refiners/process helpers incorporate knowledge on how to solve the task.

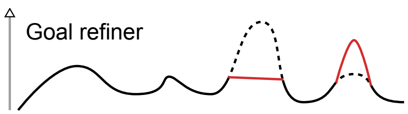
Task agnostic

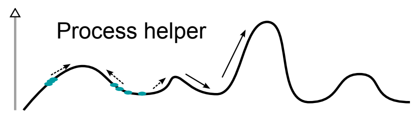
Task agnostic goal refiners/process helpers do not exploit knowledge about how to solve the task.

Doncieux, S. and Mouret J.-B. (2014).
Beyond Black-Box Optimization : a Review of Selective Pressures for Evolutionary Robotics.
Evol. Intel. DOI : 10.1007/s12065-014-0110-x, Springer Berlin Heidelberg, publisher.

Stéphane Doncieux (ISIR, UPMC-CNRS) GECCO 2015 Tutorial 9 / 25

Why modifying selective pressures ?





Goal refiners

A goal refiner aims at changing the optimum(s) of the fitness function by adding new requirements.

Process helpers

A process helper intends to increase the efficiency of the search process without changing the optimum(s) of the fitness function.

Task specific

Task specific goal refiners/process helpers incorporate knowledge on how to solve the task.

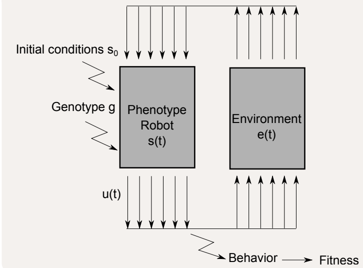
Task agnostic

Task agnostic goal refiners/process helpers do not exploit knowledge about how to solve the task.

Doncieux, S. and Mouret J.-B. (2014).
Beyond Black-Box Optimization : a Review of Selective Pressures for Evolutionary Robotics.
Evol. Intel. DOI : 10.1007/s12065-014-0110-x, Springer Berlin Heidelberg, publisher.

Evaluation

Robot interaction with the environment



$$s(t+1) = G(s(t), u(t), e(t))$$

where :

- $G(.)$ models the robot and its environment
- $s(.)$ is the state of the robot
- $u(.)$ are the control variables (motor commands)
- $e(.)$ external factors

Fitness objectives

$$f_i(g) = F_i(s_0^{(i)}, s^{(i)}(1), ..., s^{(i)}(T^{(i)}), x^{(i)})$$

where :

- $x^{(i)}$ represents other factors that the fitness objective may depend on
- $s_0^{(i)}$ is the initial state of the robot
- $T^{(i)}$ is the evaluation length

Selective pressures

Each evaluation depends on...

- the genotype g
- the fitness function $f(.)$

... but also on

- $G(.)$: robot and environment features and interaction
- $s_0^{(i)}$: the initial state of the evaluation
- $T^{(i)}$: the evaluation length
- $e^{(i)}$: the external conditions
- $s_0^{(i)}, s^{(i)}(1), ..., s^{(i)}(T^{(i)})$: the behavior of the robot

Definition

Any aspect that influences the survival or reproduction of an individual

How to modify selective pressures ?

How ? Typical task-specific approaches

Fitness shaping

“ [Shaping is] a mean to translate suggestions coming from an external trainer into an effective control strategy”

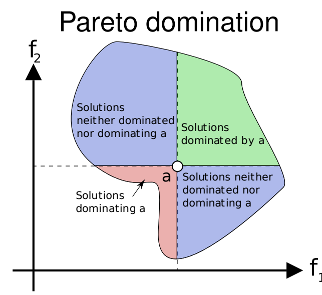
[Dorigo and Colombetti, 1994]

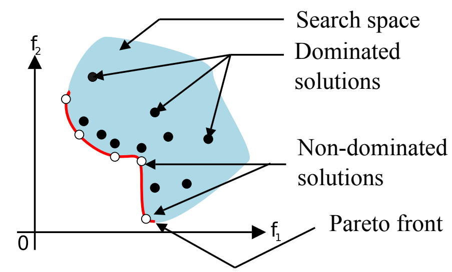
- One EA
- One fitness that includes multiple terms [Nolfi 1997]

Staged evolution

- Multiple different EA
- Examples :
  - fitness change : incremental evolution [Harvey et al. 1994, Parker 2001, ...]
  - evolution of components : modular decomposition [Urzelai et al. 1998], hierarchical evolution [Duarte et al. 2012]

How ? Multi-objective EA





Multi-objective optimization as a convenient tool to modify selective pressures

Find $g_{\mathbf{X}}(.)$ maximizing

$$\mathbf{f}(g_{\mathbf{X}}(.)) = \left\{ \begin{array}{c} f_1(\mathbf{X}) \\ f_2(\mathbf{X}) \\ \vdots \\ f_n(\mathbf{X}) \end{array} \right\}$$

At the end of the run :

- $f_i(\mathbf{X})$=Goal refiner
  → taken into account
- $f_j(\mathbf{X})$=Process helper
  → ignored

How ? Multi-objective EA

Goal refiners

Task-specific

- Evolution of internal representations [Ollion et al. 2012]

Task-agnostic

- Reality gap [Koos et al. 2013, Koos et al. 2013b]
- Generalization [Pinville et al. 2011, Lehman et al. 2013]
- Modularity [Clunes et al. 2013a]
- Diversity of solutions [Lehman and Stanley 2011]

Process helpers

Main challenge :
Premature convergence

Task-specific

- Incremental MOEA [Barlow et al. 2004]
- Staged MOEA [Mouret et al. 2006]
- ...

Task-agnostic

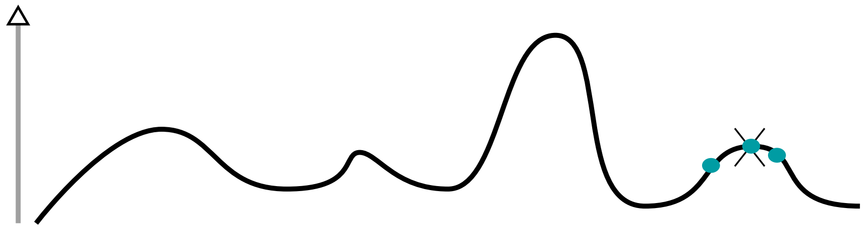
- **Behavioral diversity [Mouret and Doncieux, 2009, 2012, ...]**
- **Novelty search as an helper objective [Mouret 2011, Lehman et al. 2013, ...]**

Doncieux, S. and Mouret J.-B. (2014).
Beyond Black-Box Optimization : a Review of Selective Pressures for Evolutionary Robotics.
Evol. Intel. DOI : 10.1007/s12065-014-0110-x, Springer Berlin Heidelberg, publisher.

The challenge of premature convergence

task-agnostic process helpers

The challenge of premature convergence



Exploration vs exploitation

Intensification vs diversification in evolutionary algorithms

- Exploration : stochastic search operators & population ;
- exploitation : fitness function.

Hypothesis

Premature convergence may be due to an exploration problem.

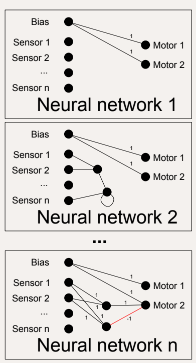
Intensification vs diversification in EA

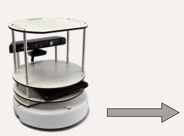
How to keep a diverse population ?

→ by penalizing similar individuals on the basis of their genotype or phenotype :

- fitness sharing [Goldberg and Richardson 1987]
- objective on diversity in a multi-objective scheme [Abbas and Deb 2003, de Jong et al. 2001]
- niches [Sareni and Krähenbühl 1998]

Exploration vs exploitation





Behavior

n neural networks → 1 behavior

Why not promoting diversity in the space of behaviors ?

[Lehman and Stanley 2008] [Trujillo et al. 2008] [Mouret and Doncieux 2009] [Gomez 2009]

Exploration vs exploitation

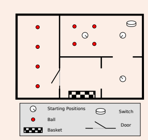
How to describe and compare behaviors ?

- adhoc descriptions :
  - final position [Lehman and Stanley 2008]
  - environment state [Mouret and Doncieux 2009]
- generic descriptions :
  - robot trajectory [Trujillo et al 2008]
  - hamming distance [Doncieux and Mouret 2010]
  - entropy [Delarboulas et al. 2011]

Behavioral diversity

Find $g_{\mathbf{X}}(.)$ maximizing $\left\{ \begin{array}{l} f(g_{\mathbf{X}}(.)) \\ \frac{1}{N}\sum_{j=0}^{j=N} d(g_{\mathbf{X}}(.), g_{\mathbf{Y}}(.)) \end{array} \right.$

with $d(\mathbf{X}, \mathbf{Y})$ behavioral distance between $\mathbf{X}$ and $\mathbf{Y}$



Mouret, J.-B. and Doncieux,S. (2012)
Encouraging Behavioral Diversity in Evolutionary Robotics : an Empirical Study
Evolutionary Computation. Vol 20 No 1 Pages 91-133.

Exploration vs exploitation

Back to the roots

The fitness function :

1. defines the goal
2. guides the search

What if a goal-oriented fitness function misguides the search ?

Doncieux, S. and Mouret J.-B. (2014).
Beyond Black-Box Optimization : a Review of Selective Pressures for Evolutionary Robotics.
Evol. Intel. DOI : 10.1007/s12065-014-0110-x, Springer Berlin Heidelberg, publisher.

Exploration vs exploitation

Novelty search

1. definition of the goal : goal-oriented fitness
2. guide for the search : novelty search

Novelty search :

1. Archive of explored behaviors
2. Fitness = Distance to the k nearest neighbors (pop+archive) :

$$\rho(x) = \frac{1}{k}\sum_{i=0}^{k} dist(x, \mu_i)$$

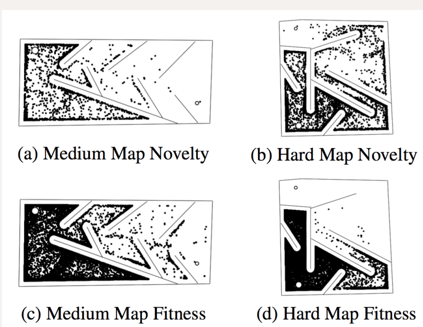
3. progressive complexification

Lehman, J., & Stanley, K. O. (2011).
Abandoning objectives : evolution through the search for novelty alone.
Evolutionary Computation, 19(2), 189–223. doi :10.1162/EVCO_a_00025.

Exploration vs exploitation

Novelty search

1. definition of the goal : goal-oriented fitness
2. guide for the search : novelty search



(a) Medium Map Novelty (b) Hard Map Novelty

(c) Medium Map Fitness (d) Hard Map Fitness

Lehman, J., & Stanley, K. O. (2011).
Abandoning objectives : evolution through the search for novelty alone.
Evolutionary Computation, 19(2), 189–223. doi :10.1162/EVCO_a_00025.

Exploration vs exploitation

Solution of the maze navigation task

After 8000 generations :

- Success/Failure only : 0/20 runs succeed
- + diversity : 1/20 runs succeed
- **+ novelty : 19/20 runs succeed**

Novelty search : to go further

- novelty as a helper objective [Mouret 2011]
- novelty and local competition [Lehman and Stanley 2011]
- novelty, optimization and interactive evolution [Wooley and Stanley 2014]
- ...

Mouret, J.-B. (2011).
Novelty-based Multiobjectivization.
In New Horizons in Evolutionary Robotics : Extended contributions of the 2009 EvoDeRob Workshop (pp. 139–154). Springer.

Lehman, J., & Stanley, K. O. (2011).
Evolving a Diversity of Creatures through Novelty Search and Local Competition.
In Proc. of the International Conference on Genetic and Evolutionary Computation (GECCO'11) (pp. 211–218).

Woolley, B. G., & Stanley, K. O. (2014).
A Novel Human-Computer Collaboration : Combining Novelty Search with Interactive Evolution.
In Proceedings of GECCO'2014.

Summary

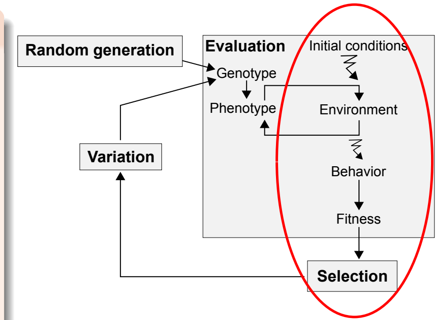
How to change sel. pressures ?

Evaluation :

- evaluation conditions
- staged evolution
- fitness shaping
- interactive evolution
- multi-objectivization

Selection algorithm :

- coevolution
- adhoc EA



Different flavors of selective pressures

- **goal refiner**
- **process helper**

→ each can be either task specific or task agnostic

Multi-objective optimization as a framework to study and define selective pressures

Doncieux, S. and Mouret J.-B. (2014).
Beyond Black-Box Optimization : a Review of Selective Pressures for Evolutionary Robotics.
Evol. Intel. DOI : 10.1007/s12065-014-0110-x, Springer Berlin Heidelberg, publisher.

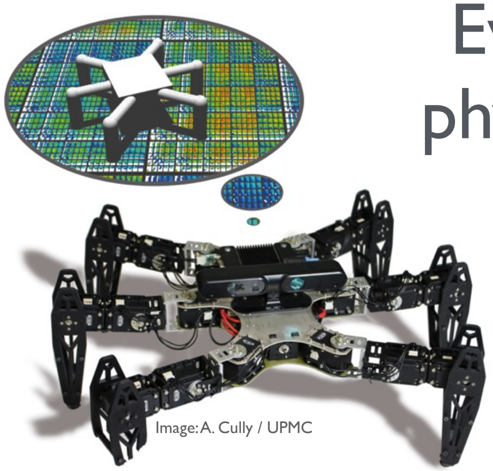
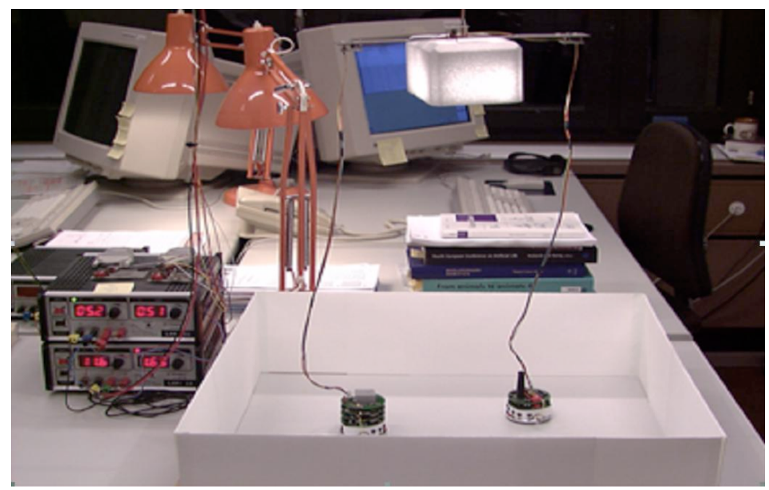


Image: A. Cully / UPMC

Evolution for physical robots

Jean-Baptiste Mouret
Inria Nancy-Grand Est

No simulator

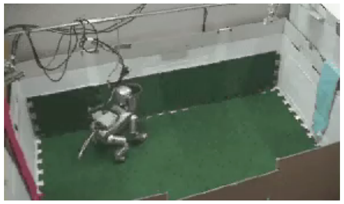


Floreano, Dario, and Francesco Mondada. "Evolution of homing navigation in a real mobile robot." Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 26.3 (1996): 396-407.

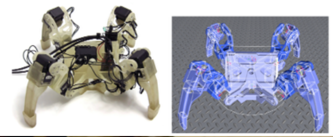
Nolfi, S., & Floreano, D. (2001). Evolutionary robotics. The biology, intelligence, and technology of self-organizing machines . MIT press.

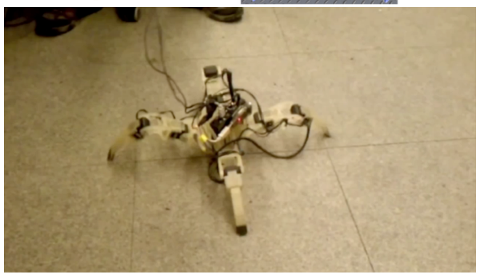
No simulator

locomotion



Hornby, G. S., Takamura, S., Yamamoto, T., & Fujita, M. (2005). Autonomous evolution of dynamic gaits with two quadruped robots. Robotics, *IEEE Transactions on*, 21(3), 402-410.





Yosinski, J., Clune, J., Hidalgo, D., Nguyen, S., Zagal, J., & Lipson, H. (2011). Evolving robot gaits in hardware: the HyperNEAT generative encoding vs. parameter optimization. In *Proc. of ECAL*, pp. 890-897.

3

No simulator

evolving walking controllers

	Starting	Time (1 run)	Robot	DOFs	Param.
Chernova and Veloso (2004)	random	5 h	quadruped	12	54
Zykov et al. (2004)	random	2 h	hexapod	12	72
Berenson et al. (2005)	random	2 h	quadruped	8	36
Hornby et al. (2005)	non-falling	25 h	quadruped	19	21
Mahdavi and Bentley (2006)	random	10 h	snake	12	1152
Barfoot et al. (2006)	random	10 h	hexapod	12	135
Yosinski et al. (2011)	random	2 h	quadruped	9	5

Pros
- (almost) no reality gap
- can exploit unknown physics

Cons
- slow (too slow?)
- will not be faster next year
- never 100% real

4

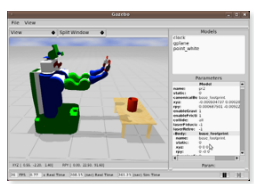
Using simulators

useful tools?

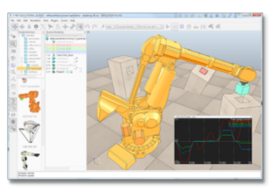
Evolution is a slow process
millions of years!

… but computers are faster every year

Can we 'accelerate time'?



We now have many "good" simulators:
- ODE (library): www.ode.org
- Bullets (library): bulletphysics.org
- Gazebo (GUI): gazebosim.org
- V-Rep (GUI): www.coppeliarobotics.com
- …



Tools for dynamics simulation of robots: a survey based on user feedback. S Ivaldi, V Padois, F Nori. *Proc. of Humanoids 2014*.

5

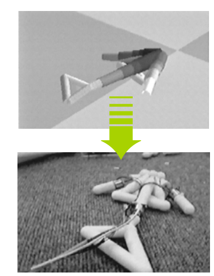
Automatic design

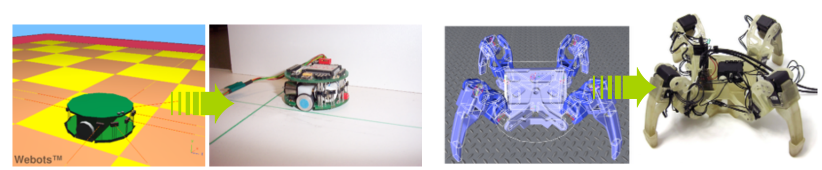
the ideal process

1. Evolve in simulation

2. Transfer the result to the reality
   ➠ build the robot
   ➠ upload the controller to the robot

3. Enjoy your optimal design / controller

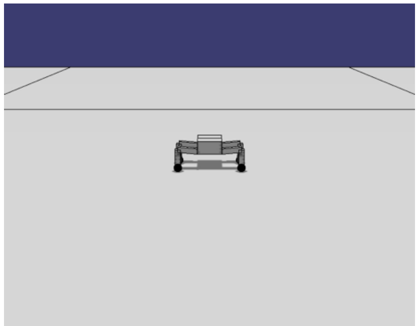


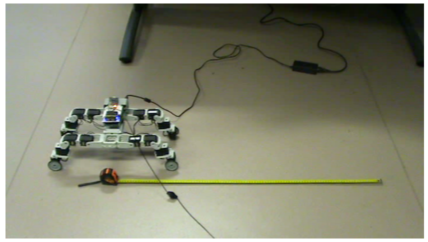


6

The reality gap

… or what always happens in evolutionary robotics



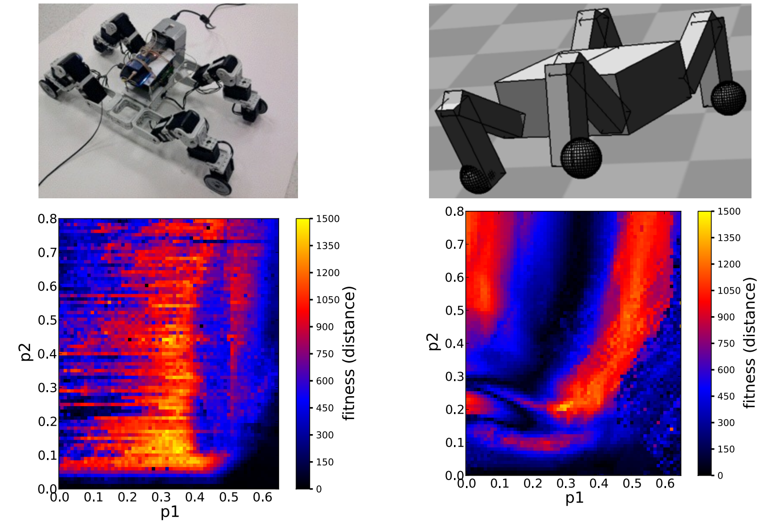


Koos, Mouret & Doncieux.
IEEE Transactions on Evolutionary Computation.
2012

Controller: 2 parameters

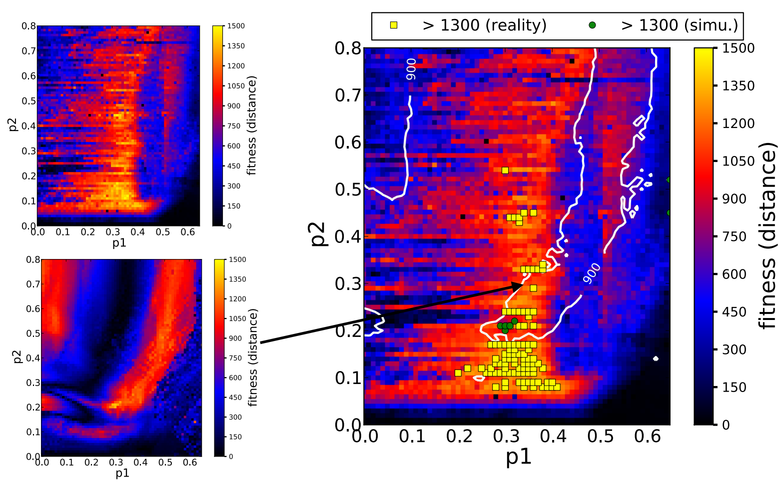
Jakobi, Nick. "Running across the reality gap: Octopod locomotion evolved in a minimal simulation." Evolutionary Robotics. Springer Berlin Heidelberg, 1998.

Reality vs simulation



Mouret, J. B., Koos, S., & Doncieux, S. (2013). Crossing the reality gap: a short introduction to the transferability approach. arXiv preprint arXiv:1307.1870.

But they can agree (sometimes)



The reality gap

- Any simulation has a validity domain
- Human experts know this validity domain
- … but evolution does not have this common sense

Results found in simulation are not likely to work similarly in reality

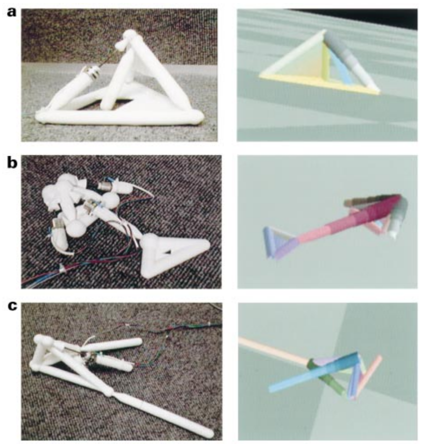
➠ **One of the main challenge of ER**

What can we do?

no simulator
better simulator
avoid non-transferable solutions
robust controllers

Finish evolution in reality

evolve in simulation, then do a few generations with the robot



Pro: can help fine-tuning the solution obtained in simulation

Con: "local search" in the vicinity of the solutions found in simulation

➠ cannot find something completely different

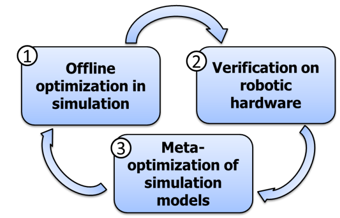
Lipson, H., & Pollack, J. B. (2000). Automatic design and manufacture of robotic lifeforms. **Nature**, 406, 974–978.

Improving simulators

General idea: minimize the difference between simulation and reality (supervised learning)

➠ Miglino et al.: measure the exact response of the infrared sensors (Khepera)

➠ Moeckel et al.: optimize the parameters of an ODE simulator (22 parameters) ; PSO

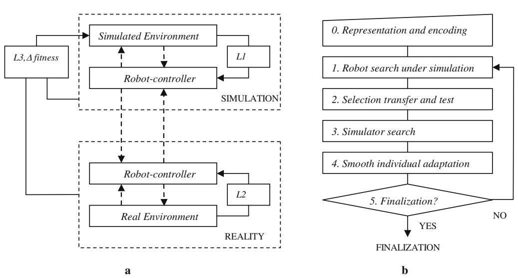


Miglino et al. "Evolving mobile robots in simulated and real environments." Artificial life 2.4 (1995): 417-434.

Moeckel et al. "Gait optimization for roombots modular robots—Matching simulation and reality." Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on (IROS), 2013.

Improving simulators

the "back to reality" algorithm

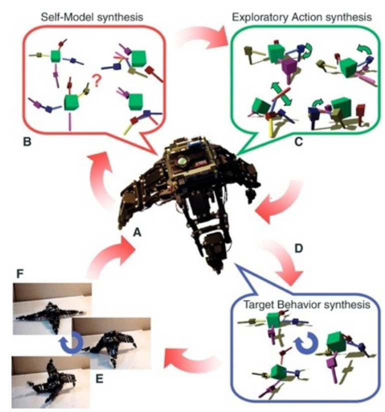


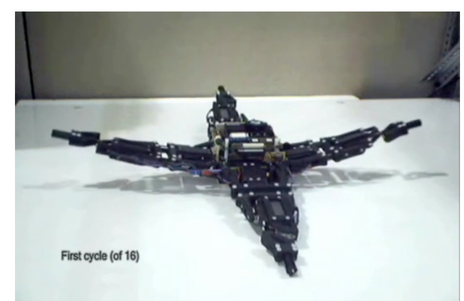
L1, L2 and L3: evolutionary algorithms
200 evals on the robot, 10 500 simulation
compare fitness values

Zagal, Juan Cristóbal, and Javier Ruiz-Del-Solar. "Combining simulation and reality in evolutionary robotics." Journal of Intelligent and Robotic Systems 50.1 (2007): 19-39.

Improving simulators

The EEA algorithm: active learning of a model





Bongard, Zykov and Lipson. Science. 2006

Koos, S., Mouret, JB and Doncieux, S.. "Automatic system identification based on coevolution of models and tests." Evolutionary Computation, 2009. CEC'09. IEEE Congress on. IEEE, 2009.

Improving simulators

Pros	Cons
mix simulation and reality: the best of both worlds?	the simulator will never be perfect
faster than learning without a simulator	if the correction cannot be applied? (e.g. aerodynamics)
morphological / env. changes	learning a simulator is hard!

Related work: optimization with surrogate fitness functions (learn a "simulator" from scratch)

Jin, Yaochu. "A comprehensive survey of fitness approximation in evolutionary computation." Soft computing 9.1 (2005): 3-12.

15

Avoiding bad simulations

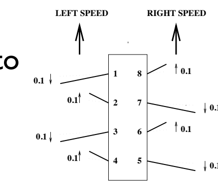
the envelope of noise & minimal simulations

Simulate only the useful effects
Hide in an "envelope of noise" things that are too hard to simulate accurately

- ➠ keep evolution from exploiting simulation artefacts
- ➠ goal refiner

Examples:

- ➠ Khepera robot: add noise to the sensors and the actuators
- ➠ Octopod robot: minimal simulation



Jakobi, N. "Evolutionary robotics and the radical envelope-of-noise hypothesis." Adaptive Behavior 6.2 (1997): 325-368.

16

Avoiding bad simulations

envelope of noise & minimal simulations

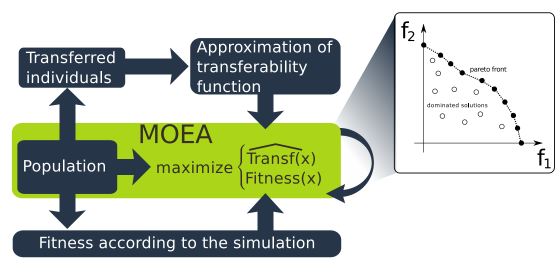
Pros	Cons
Lightweight simulations	Hard to set-up
Noise increases robustness and generalization	What noise? what is important?
	No surprising dynamic effect
	Noise makes evolution harder

17

Avoiding bad simulations

the transferability approach

- ➠ learn the limits of the simulation (supervised learning)
- ➠ focus the search on well-simulated behaviors
- ➠ the transferability is a task-agnostic goal refiner



Mouret, Koos & Doncieux. ALIFE workshop. 2012
Koos, Mouret & Doncieux. IEEE TEC. 2012
Koos, Cully & Mouret. IJRR. 2013

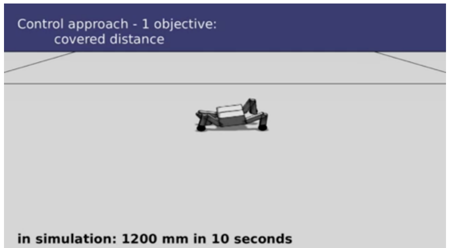
18

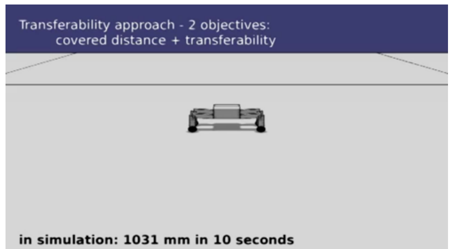
Avoiding bad simulations

the transferability approach

Maximize fitness

Maximize { fitness, transferability }



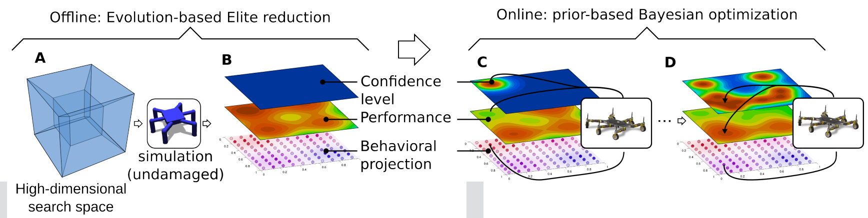


15 transfers
(motion capture)

Koos, S., Mouret, J.-B., & Doncieux, S. (2011). The Transferability Approach : Crossing the Reality Gap in Evolutionary Robotics. IEEE Transaction on Evolutionary Computation, 1, 1–25.

Mapping, then searching

Intelligent Trial & Error



Mapping (offline)
- MAP-Elites algorithm
- search for the best behavior of each family

Adaptation (online)
- Bayesian optimization
- levels of confidence

Cully, Clune, Tarapore & Mouret. Nature. 2015

Mapping, then searching

Intelligent Trial & Error

Cully, Clune, Tarapore & Mouret. Nature. 2015

Avoiding bad simulations

the transferability approach

Pros

Easier to learn the limit than to correct/learn the simulator

Only a few test on the robot: no need for a special set-up

Cons

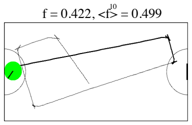
The EA cannot exploit phenomena that not simulated at all

(e.g. highly-dynamic gaits, unknown aerodynamic effects, etc.)

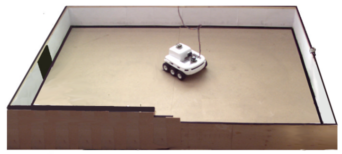
Improving robustness

evolve controllers with online learning abilities

Example: neural networks with "adaptives synapses"



Adaptive synapses



Floreano, D., & Urzelai, J. (n.d.). Evolutionary robots with on-line self-organization and behavioral fitness. Neural Networks, 13(4-5), 431–43.

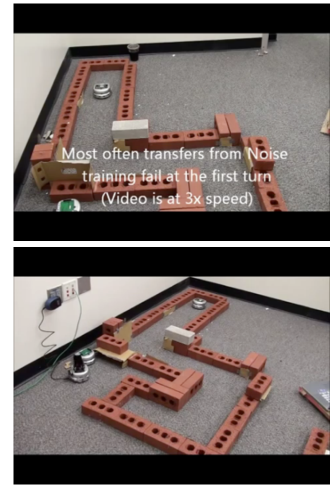
Urzelai, J., & Floreano, D. "Evolutionary robots with fast adaptive behavior in new environments." Evolvable Systems: From Biology to Hardware. Springer Berlin Heidelberg, 2000. 241-251.

23

Improving robustness

encouraging reactivity

- quantification of reactivity derived from the mutual information between sensors and actuators
- multi-objective optimization
- even better if combined with noise
- task-agnostic goal refiner



Lehman, Joel, et al. "Encouraging reactivity to create robust machines." Adaptive Behavior (2013): 1059712313487390.

24

Conclusion

the reality gap

1	➠ No simulator: possible but slow
2	➠ Finish evolution on the physical root: similar optima
3	➠ Improving simulators: *cannot learn everything* ▸ *EEA, back to reality, …*
4	➠ Avoiding badly simulated solutions (goal refiners) ▸ *add noise to sensors and actuators: hard to tune* ▸ *minimal simulations: requires expert knowledge* ▸ *learn the transferability function*
5	➠ Improving robustness (goal refiners): no guarantee ▸ *add online learning abilities* ▸ *encourage reactivity*

25

GECCO 2015, Madrid

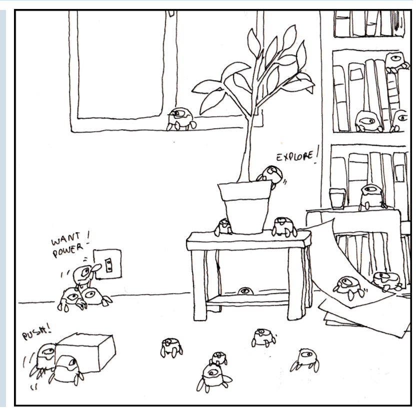
Evolutionary robotics and collective adaptive systems

Tutorial « Evolutionary Robotics », part 3/3

July 2015

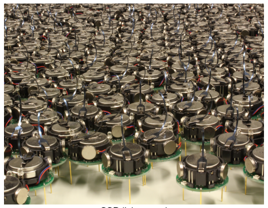
Nicolas Bredeche

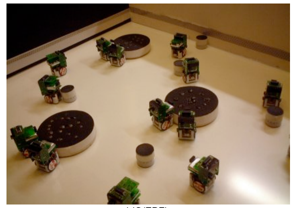
Université Pierre et Marie Curie
Institut des Systèmes Intelligents et de Robotique
ISIR, UMR 7222
Paris, France
nicolas.bredeche@upmc.fr



Note on citation policy: for a given topic, I cite either or both the seminal reference and a recent one. E.g.: [Nolfi and Floreano. 2000][Doncieux et al. 2015] for referring to general resources on evolutionary robotics. Non-first authors may be omitted for clarity.


Kiva/Amazon


SSR/Harvard


LIS/EPFL

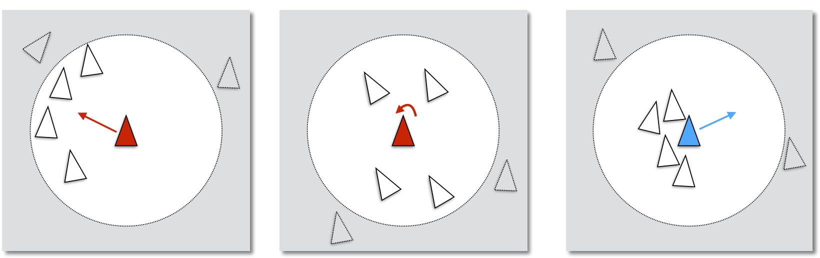
Collective robotics: multiple robots, acting together, to achieve a common goal.

Swarm robotics: collective robotics with large population of "simple" robots (i.e. *limited computation and communication capabilities*). It is a *distributed* system.

In this talk:

we focus on <u>distributed</u> robotic systems, with <u>small or large</u> groups

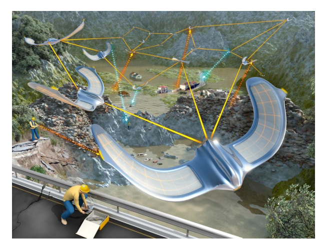
limited computation and communication capabilities



Attraction Orientation Repulsion

Positive and negative feedbacks
- positive feedback: *attraction and orientation rules*
- negative feedback: *repulsion rule*

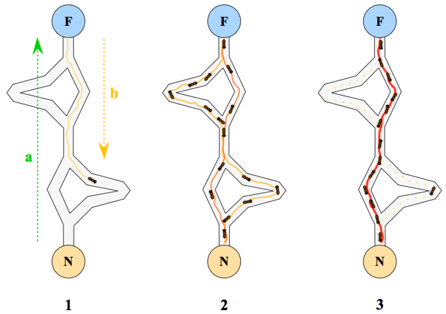
Physical structure	homogeneous		
Control			distributed
Control design			optimised
Control at run-time		fixed	



Auton Robot (2009) 26: 21–32
DOI 10.1007/s10514-008-9104-9

Evolved swarming without positioning information: an application in aerial communication relay

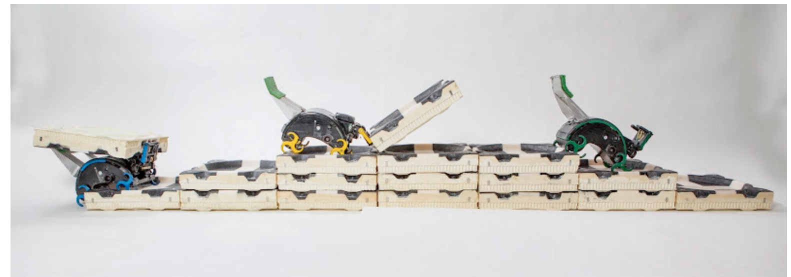
Sabine Hauert · Jean-Christophe Zufferey · Dario Floreano



Stigmergy: indirect coordination between agents through a (chemical or physical) element left in a shared environment.
e.g.: *pheromones, obstacles*

Dorigo et al. (1996) Ant system: optimization by a colony of cooperating agents

Physical structure	homogeneous
Control	distributed
Control design	optimised
Control at run-time	fixed



Science AAAS

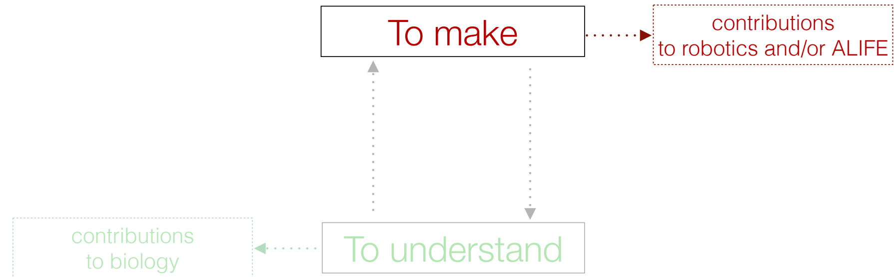
Designing Collective Behavior in a Termite-Inspired Robot Construction Team
Justin Werfel *et al.*
Science **343**, 754 (2014);
DOI: 10.1126/science.1245842

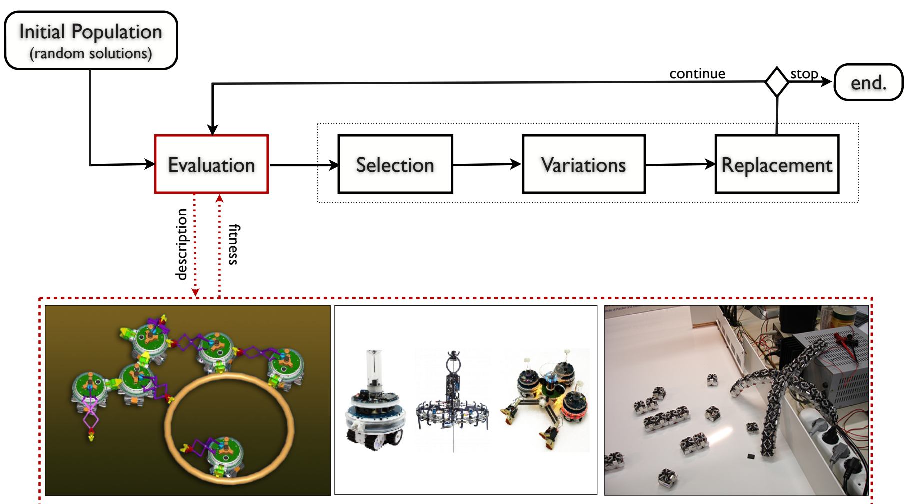
- Defining the problem [Nettleton et al., 2003], adapted from [Capitan et al. 2013]
  - no central control
  - no common communication facility
  - no local knowledge of the team global topology
- Obvious advantages
  - Robustness through redundancy
  - Parallelising actions wrt a task
  - Parallelising learning/optimisation (if any)
- Critical problems
  - Solving distributed decision problems is NEXP-complete (if exact sol.)
  - Even approximated methods provides limited results (few robots in practical)
  - Predicting the outcome of simples rules is challenging (complex dynamics)

- Approaches
  - hand-coded
    - (Trial&error) top-down approach [Mataric, 1992+][McLurkin, 2004+][...]
    - (Bio-inspired) bottom-up approach [Bonabeau et al., 1999 for an introduction][Reynolds, 1984][...]
  - learning and optimisation
    - Brute force optimisation [Werfel et al., 2014][...]
    - Exact and approximate method in RL [Bernstein,2002][Amato, 2014][...]
    - Evolutionary algorithm (meta-heuristic for policy search)
      - continuous states and actions
      - non-standard representations
      - not just optimisation (cf. earlier presentations)
      - versatile wrt. collective setups (clones, non-clones, structured populations, etc.

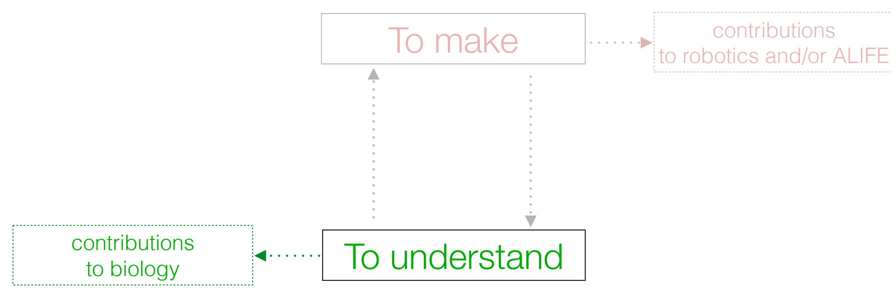
Evolving collective robotics :

Dual motivations

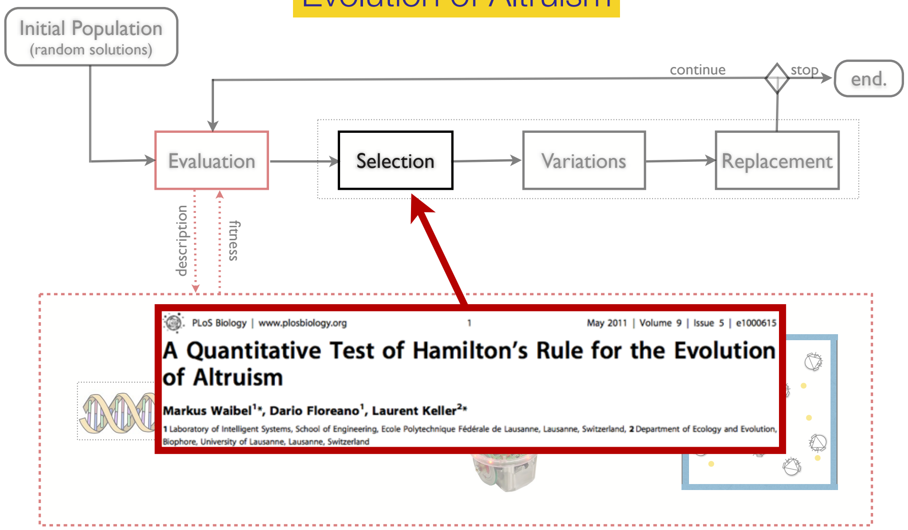




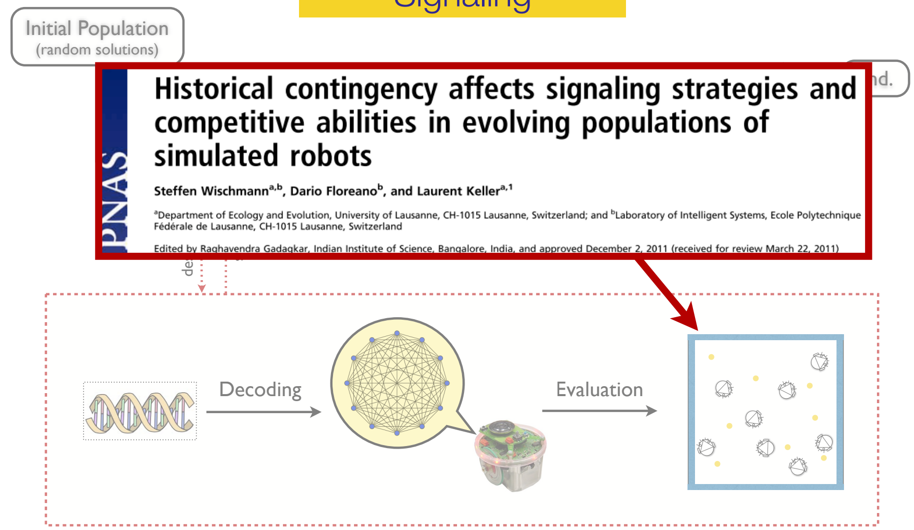
Swarm-bots, 2001-2005 Swarmanoid, 2006-2010 Symbrion and Replicator, 2008-2013



Evolution of Altruism



Signaling

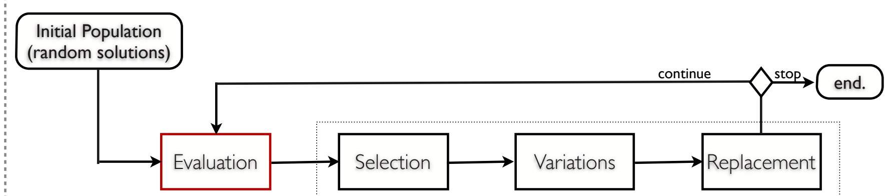


Evolving collective robotics :

Dual methods

Dual methods

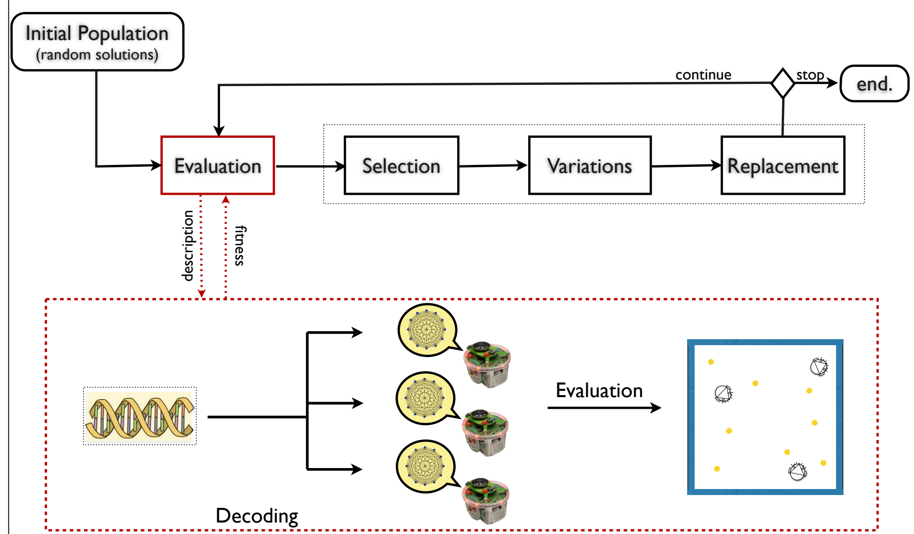
(Off-line) classic evolutionary robotics [Nolfi, Floreano 2000][Doncieux et al. 2015]

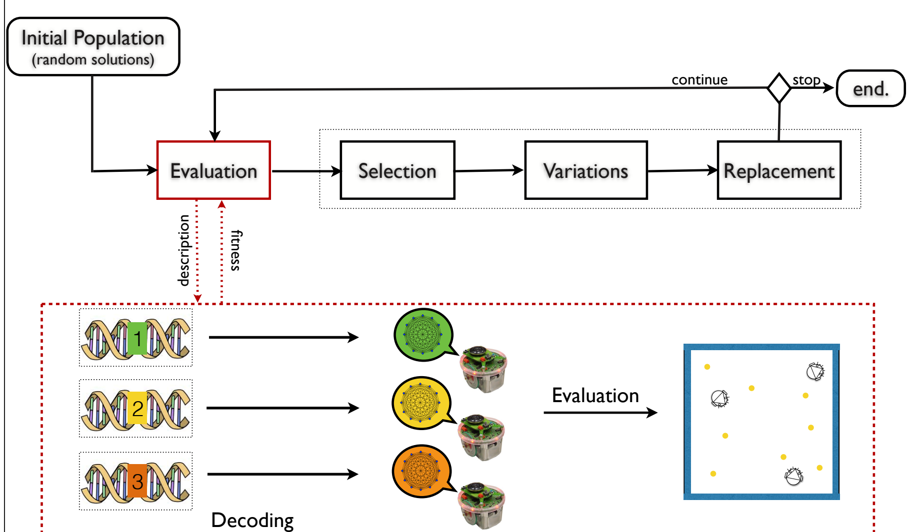


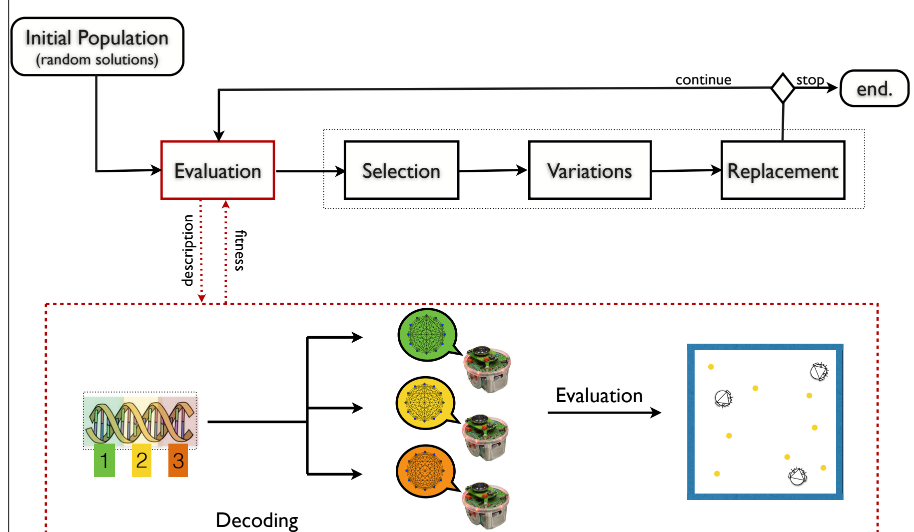
(On-line) embodied evolution [Watson et al. 2002][Eiben et al. 2010]

Focus	control	*and/or*	morphology
Space	centralized	*or*	distributed
Time	off-line	*or*	on-line
Selection pressure	fitness function	*and/or*	environment-driven

- What?
  - Off-line design method
  - Optimize in centralized fashion, then used in a distributed fashion
- Expected result
  - A set of policies (*possibly similar*) that can be used within a population of robots to solve a task

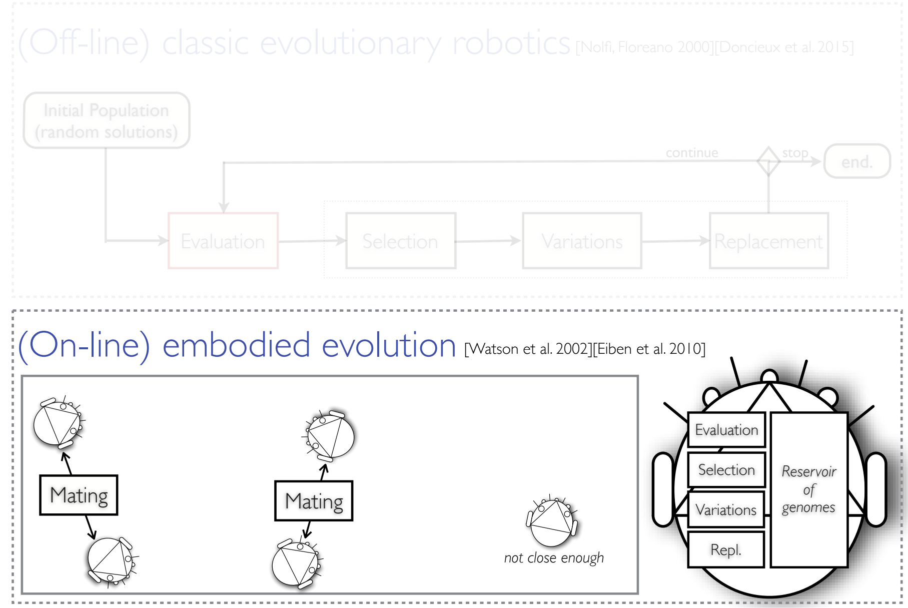






- Lessons learned
  - Cooperation
    - between related individuals (inclusive fitnesses) [Waibel 2009,2011]
    - between unrelated individuals (mutualism) [Bernard 2015]
  - Communication and signalling
    - efficiency vs. robustness [Wischmann 2012]
    - directional communication [Pugh 2014]
- Open issues (in addition to other classical issues with ER)
  - Improving on scalability *and* complexity
  - Division of labour
    - Behaviour heterogeneity in homogeneous populations [D'Ambrosio 2013]
    - Cooperative behaviours in heterogeneous population (e.g. mutual adaptation [Ducatelle 2010])

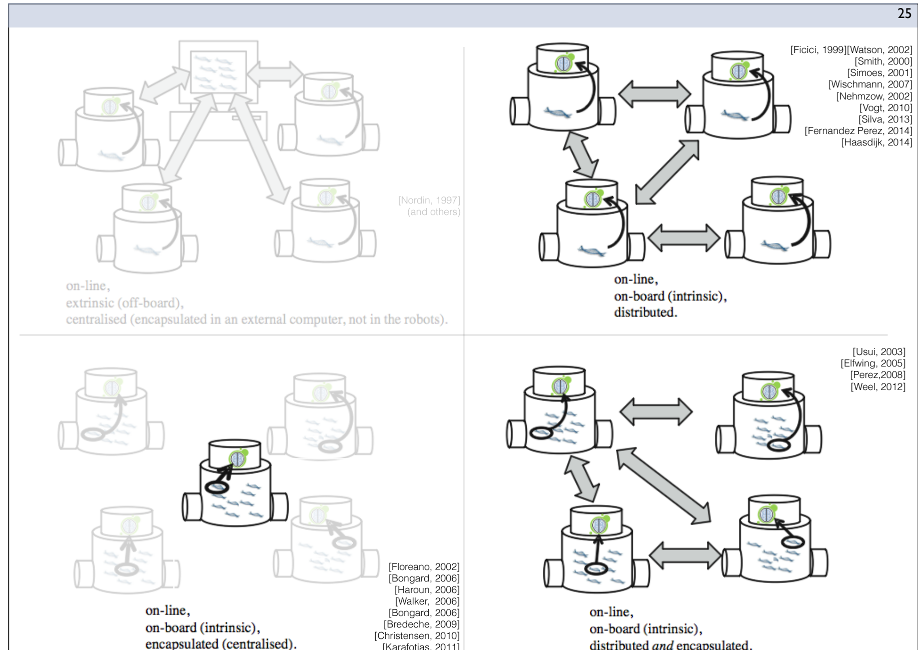
Dual methods



Definitions

Focus	control	and/or	morphology
Space	centralized	or	distributed
Time	off-line	or	on-line
Selection pressure	fitness function	or	environment-driven

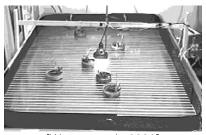
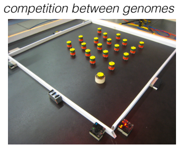
- What?
  - On-line adaptation
  - Optimised and used in a distributed fashion
- Expected result
  - A population of robots improving over time wrt. a task to achieve
  - Continuous adaptation to open, possibly changing, environments



 [Eiben, Haasdijk, Bredeche, 2010]

directed evolution... ...natural evolution

Typical examples (non-exhaustive)

Fitness function selection depends from a user-defined metric	**Environment-driven** selection results from interactions in the environment
directed selection task-explicit: *patrolling, reach a goal, etc.* task-implicit: *energy-driven, predator-prey, etc.* task-free: *diversity, novelty*	natural selection ***no** fitness function* *the resulting behaviour is shaped by the environment and the competition between genomes*
	
[Watson et al., 2002] [Eiben et al., 2010] [Trueba et al. 2012] etc.	[Bredeche and Montanier. 2010] [Bianco et al., 2004] etc.

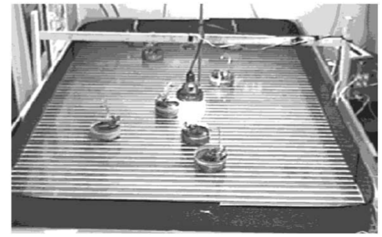


Fig. 4. The robot pen for the phototaxis experiments. Eight robots, the power floor, and the light in the center are shown. The unique ID of a robot is collected when it reaches the light (via infrared receivers on the overhead beam above the lamp). This data is time-stamped and stored for monitoring experiment progress.

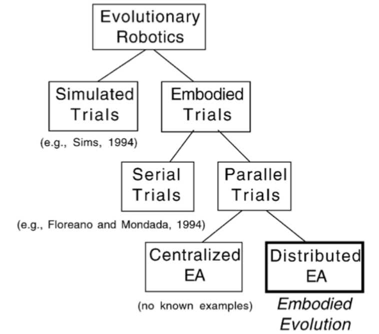


Fig. 1. Embodied Evolution is an evolutionary robotics methodology that embodies a distributed evolutionary algorithm within a population of real robots.

Embodied Evolution: Distributing an evolutionary algorithm in a population of robots
Richard A. Watson, Sevan G. Ficici, Jordan B. Pollack
Robotics and Autonomous Systems 39 (2002) 1-18

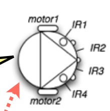
Embodied evolution

A vanilla algorithm

Controller

e.g.: weighted combination of inputs, artificial neural networks, etc.

motor1 = a*IR$_1$ + b*IR$_2$ + c*IR$_3$ + d*IR$_4$ + e
motor2 = f*IR$_1$ + g*IR$_2$ + h*IR$_3$ + i*IR$_4$ + j

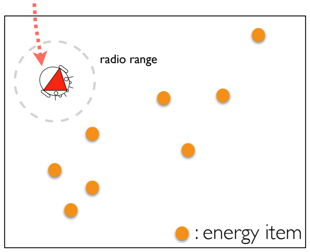


IR = Infrared sensors

Genome

<
a = +0.31
b = +0.11
c = -1.42
d = +1.6
e = -0.14
f = 0.55
g = -1.17
h = +0.97
>

genome of controller
e.g.: $\mathbb{R}^n$



Reservoir of genomes

{genome,fitness value}^n

This list is used to store (unique) **copies of genomes** from robots passing nearby with their **current fitness value** at the time of encounter

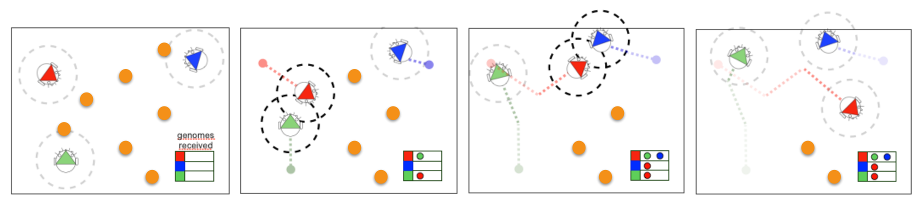
Fitness function

E.g.: #energy items foraged

A robot is «dead» if...
- internal cause: no genome available when a new generation starts
- external causes: failures, crashes, lack of energy, etc.

"Vanilla" embodied evolution algorithm

example with a foraging task



Step 1 : generation starts (3 robots, empty lists)
Step 2
Step 3
Step 4 : end of generation

At this point, each robot...

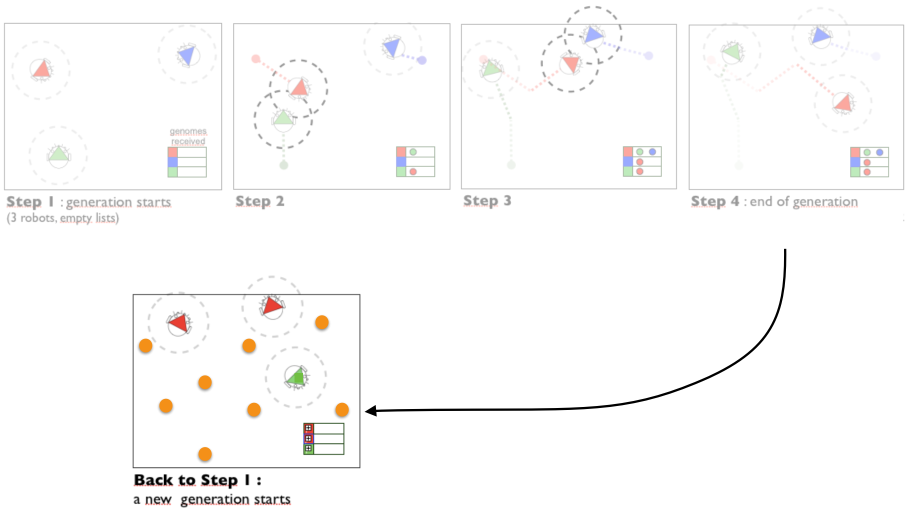
1 - **forgets** its own genome

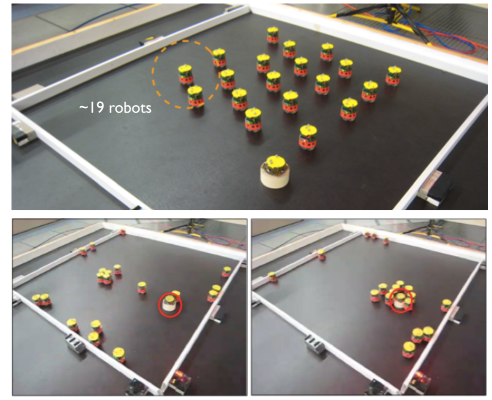
2 - perform selection among **received** genomes wrt fitness values

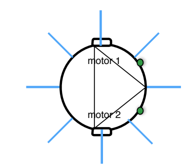
3 - apply **variation** (crossover and/or mutation) on the selected genome (e.g. gaussian mutation)

4 - use new genome to set up new control architecture

Selection pressure is applied at the **individual** level example with a foraging task







inputs:
- 8 IR sensors
- 8 bumpets
- orientation wrt. landmark
- distance to landmark

outputs:
- left and right motor speed

Mathematical and Computer Modelling of Dynamical Systems 2012

Environment-driven Distributed Evolutionary Adaptation in a Population of Autonomous Robotic Agents

Nicolas Bredeche[a,*], Jean-Marc Montanier[a], Wenguo Liu[b] and Alan F.T. Winfield[b]

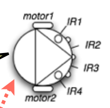
Embodied evolution

A vanilla algorithm

Controller

e.g.: weighted combination of inputs, artificial neural networks, etc.

motor1 = a*IR$_1$ + b*IR$_2$ + c*IR$_3$ + d*IR$_4$ + e
motor2 = f*IR$_1$ + g*IR$_2$ + h*IR$_3$ + i*IR$_4$ + j



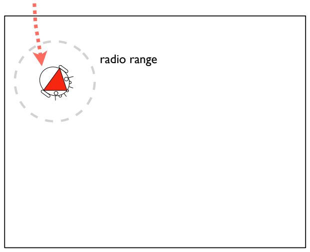
IR = Infrared sensors

Genome

<
a = +0.31
b = +0.11
c = -1.42
d = +1.6
e = -0.14
f = 0.55
g = -1.17
h = +0.97
>

genome of controller

e.g.: $\mathbb{R}^n$



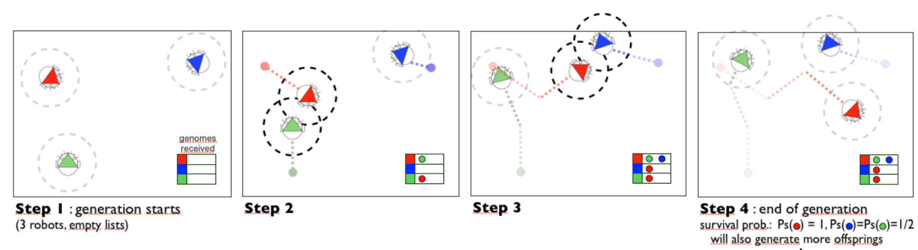
Reservoir of genomes

{genome}^n

This list is used to store (unique) **copies of genomes** from robots passing nearby.

NO Fitness function

no fitness value is computed, therefore none is stored



At this point, each robot...

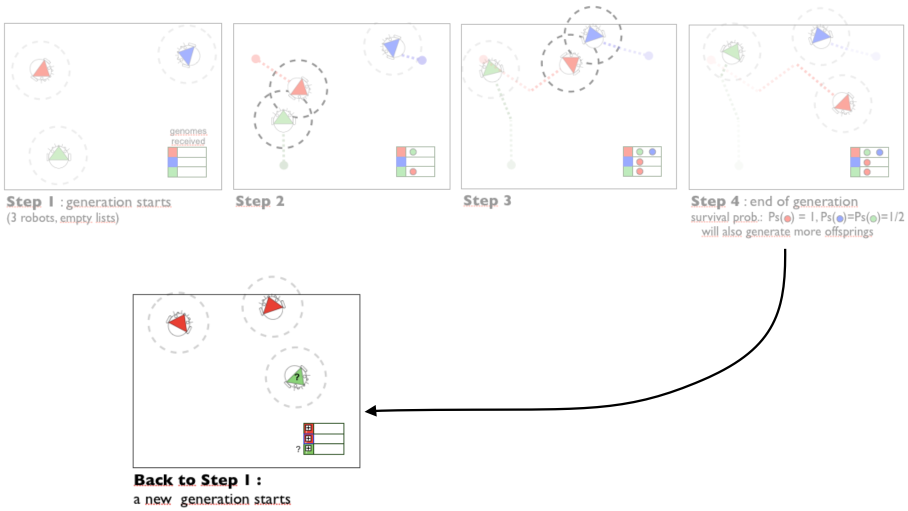
1 - **forgets** its own genome

2 - perform **random** selection among **received** genomes

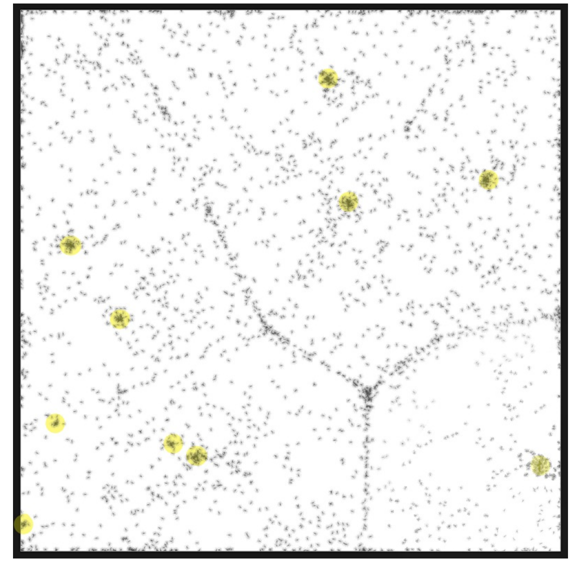
3 - apply a slight **variation** on the selected genome (e.g. gaussian mutation)

4 - use new genome to set up new control architecture

Selection pressure is applied at the **population** level



4000 robots in simulation

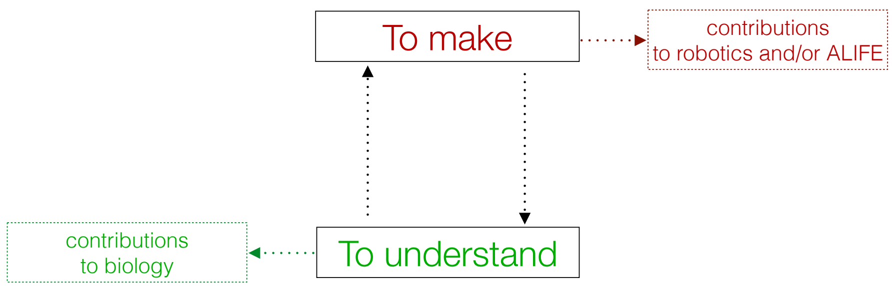


- Lessons learned
  - No reality gap (by definition)
  - Scalable algorithms (by definition)
  - Population density and communication range are critical
  - Natural evolution can be simulated — relevance to evol. biology
- Open issues
  - Specialisation is challenging [Trueba et al., 2013]
  - Evolving complex social behaviours (cooperation, division of labour, ...)
  - Evaluation/maturation time [Wischmann et al., 2007][Bredeche et al., 2009]
  - Trade-off between addressing a task and surviving [Haasdijk et al., 2014]
  - Necessary conditions for truly open-ended evolution [Bedau et al. 2000]

Wrapping up

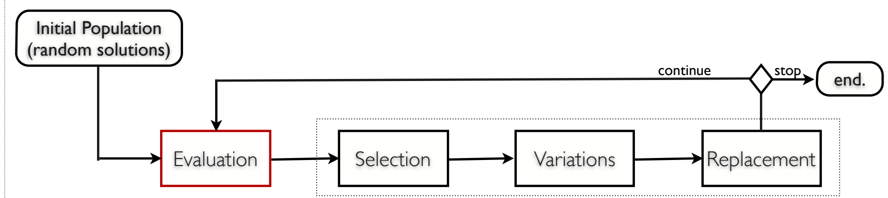
Conclusions and open issues

Dual motivations

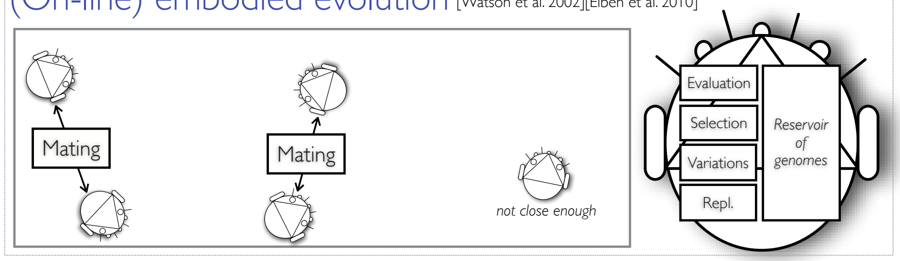


Dual methods

(Off-line) classic evolutionary robotics [Nolfi, Floreano 2000][Doncieux et al. 2015]



(On-line) embodied evolution [Watson et al. 2002][Eiben et al. 2010]



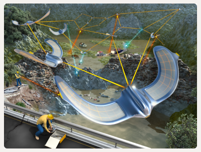
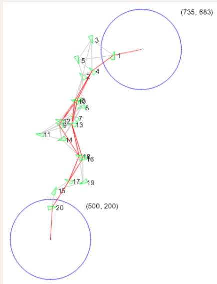
Thank you for your attention

Mailing list: https://www.listes.upmc.fr/wws/review/roborobo

Open issues

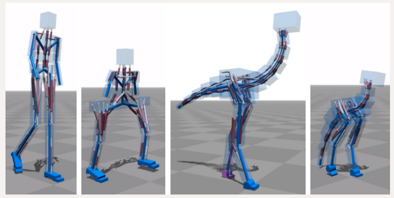
Applying ER to real world problems

ER is a successful building block

Projet SMAVNET, LIS, EPFL.

ER leads to impressive results in simulation



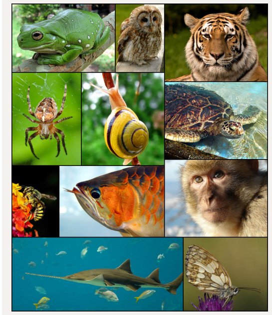
Geijtenbeek, T. and van de Panne, M. and van der Stappen, A. F. (2013)
Flexible Muscle-Based Locomotion for Bipedal Creatures.
ACM Transactions on Graphics (Proc. SIGGRAPH ASIA 2013).

... but the holistic approach of ER on real robots remains a challenge :

- How to deal with large number of evaluations ?
- How to take the best of simulation and reality ?

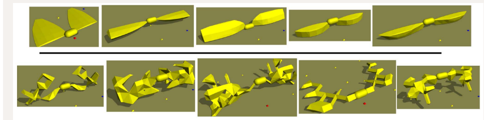
Nature-like evolvability

Natural evolution

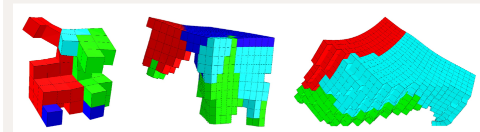


author : `https://en.wikipedia.org/wiki/User:Justin`

Artificial evolution



[Auerbach and Bongard 2014]



[Cheney et al. 2013]

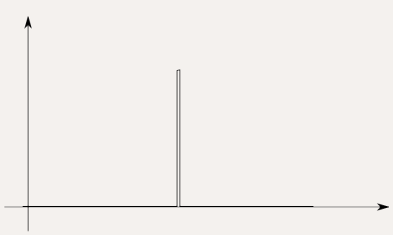
What genotype for complex systems ?
How to define **viable mutations** ?

Combining evolution and learning

Evolution and learning occur in many species, what about robots ?

Complementary in theory...

Learning can smooth a search landscape [Baldwin 1896].



Baldwin effect
[Hinton and Nowlan 1987]

... but hard to use in practice

A few hints :

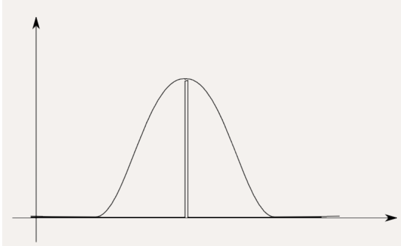
- learning to learn is deceptive [Risi et al. 2009, 2010]
- regularity in network structure makes a difference [Tonelli and Mouret 2013]
- needs formalization [Mouret and Tonelli 2014]

Combining evolution and learning

Evolution and learning occur in many species, what about robots ?

Complementary in theory...

Learning can smooth a search landscape [Baldwin 1896].



Baldwin effect
[Hinton and Nowlan 1987]

... but hard to use in practice

A few hints :

- learning to learn is deceptive [Risi et al. 2009, 2010]
- regularity in network structure makes a difference [Tonelli and Mouret 2013]
- needs formalization [Mouret and Tonelli 2014]

Evolutionary robotics and reinforcement learning

Convergence of the two approaches ...

... for policy optimization in continuous spaces :

- ER similar to policy search algorithms [Kober et al. 2013]
- ER competitive with recent RL algorithms [Stulp and Sigaud 2012]

... but not for all ER applications :

- morphology design is out of the scope of RL
- RL is not a model of biological evolution

- What inspiration to draw from RL ?
- What new algorithms to build for RL based on ER principles ?

Online learning : single and multiple robots

Most ER works deal with off-line learning

- evaluation in the same initial conditions
- ... for the same period
- ... and in a constant environment

Open issues with on-line learning

How to deal with :

- changing initial conditions
- changing environments

From single to multiple robots

Finding exact solutions to a multiple robots setup (DEC-POMDP) is NEXP-Complete [Papadimitriou 1994].

ER allows from several dozens of real robots [Watson et al. 2002, Bredeche et al. 2012] or thousands of simulated ones [Bredeche 2014].

Environment-driven evolutionary robotics

The limits of goal-driven search

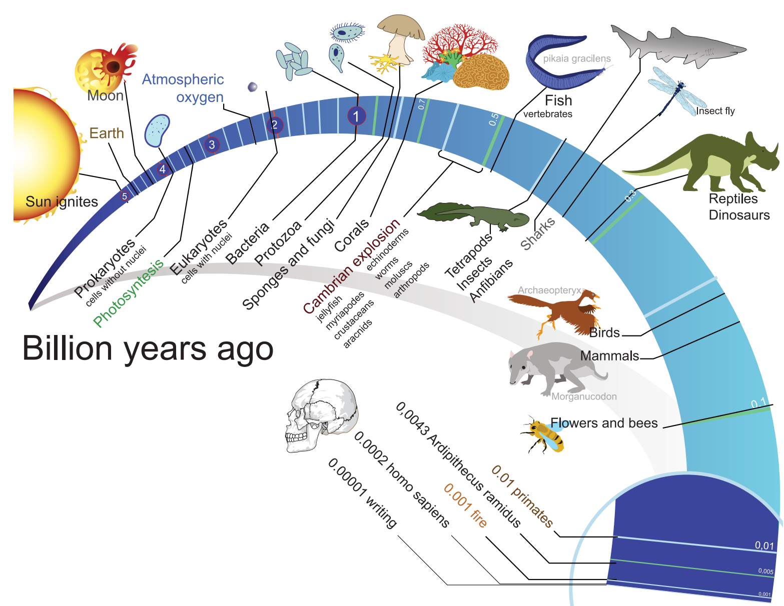
- Goal driven objectives are often deceptive [Lehman and Stanley 2011]
- What fitness function and evaluation conditions to evolve life-like capabilities ?

An alternative...

No fitness function : environment-driven evolution [Bianco and Nolfi 2004, Montanier and Bredeche 2010, Bredeche et al. 2012].

How does tasks and environment driven pressure interact ? [Haasdijk et al. 2014]

Open-ended evolution



Acknowledgements

DREAM project, FET ProActive H2020 (2015-2018)

Goal : enable robots to gain an open-ended understanding of the world over long periods of time

Main ideas :

- evolutionary approach to development
- alternating between
  - active interaction (daytime) and
  - passive introspection over past events (nighttime and dreams)

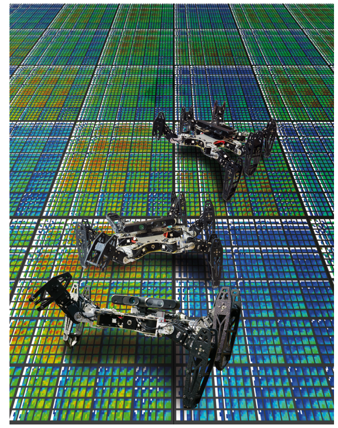
http://www.robotsthatdream.eu

https://twitter.com/robotsthatdream

ENSTA ParisTech · ARMINES · Queen Mary University of London · vrije Universiteit amsterdam · UPMC SORBONNE UNIVERSITÉS · cnrs · UNIVERSIDADE DA CORUÑA

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 640891.

Acknowledgements



ANR Creadapt (2013-2015)

Goal : use evolution as a creative adaptation tool.

- better evolvability (modularity, GDS)
- evolution on physical robots thanks to the transferability approach
- application to damage recovery

ERC ResiBots (2015-20120)

Goal : fast but creative learning algorithms for damage recovery in robotics

- EA + Bayesian Opt. (IT&E)
- multi-task & multi-objective learning
- application to damage recovery

UPMC SORBONNE UNIVERSITÉS · UNIVERSITÉ DE LORRAINE · erc · ANR · cnrs · Inria

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 637972 (ERC ResiBots) and from the Agence Nationale de la Recherche (Creadapt project, ANR-12-JS03-0009).

Questions ?

nicolas.bredeche@upmc.fr
stephane.doncieux@upmc.fr
jean-baptiste.mouret@inria.fr

Doncieux S, Bredeche N, Mouret J-B and Eiben AE (2015).
Evolutionary robotics : what, why, and where to.
Front. Robot. AI 2 :4. doi : 10.3389/frobt.2015.00004.

References : recent reviews & introduction papers

Doncieux S, Bredeche N, Mouret J-B & Eiben AE (2015).
Evolutionary robotics : what, why, and where to.
Front. Robot. AI 2 :4. doi : 10.3389/frobt.2015.00004.

Doncieux, S. and Mouret, J.-B. (2014).
Beyond black-box optimization : a review of selective pressures for evolutionary robotics.
Evolutionary Intelligence, Springer Berlin Heidelberg, publisher. Vol 7 No 2 Pages 71-93.

Bongard, J. C. (2013)
Evolutionary robotics
Communications of the ACM 56.8 : 74-83

Floreano, D., Dürr, P., & Mattiussi, C. (2008).
Neuroevolution : from architectures to learning. Evolutionary Intelligence, 1(1), 47–62.
Evolutionary Intelligence, 1(1), 47–62.

References : landmark papers

Lehman, J., and Kenneth O. Stanley (2011).
Abandoning objectives : Evolution through the search for novelty alone.
Evolutionary computation 19.2 (2011) : 189-223.

Bongard, J., V. Zykov, and H. Lipson (2006).
Resilient machines through continuous self-modeling.
Science 314.5802 :1118-1121.

Hornby, G. S., S. Takamura, T. Yamamoto, and M. Fujita (2005).
Autonomous evolution of dynamic gaits with two quadruped robots.
IEEE Transactions on Robotics, 21, no. 3 : 402-410.

Watson, R. A., S. G. Ficici, and J. B. Pollack. (2002)
Embodied evolution : Distributing an evolutionary algorithm in a population of robots.
Robotics and Autonomous Systems 39, no. 1 : 1-18.

Lipson, H., and J B. Pollack. (2000)
Automatic design and manufacture of robotic lifeforms.
Nature 406.6799 : 974-978.

Floreano, D., and F. Mondada. (1996)
Evolution of homing navigation in a real mobile robot.
IEEE Transactions on Systems, Man, and Cybernetics, Part B : Cybernetics 26.3 (1996) : 396-407.