#### **Evolutionary Robotics Tutorial**

N. Bredeche, S. Doncieux

J.-B. Mouret

Sorbonne Universités, UPMC Univ Paris 06, CNRS, Institute of Intelligent Systems and Robotics (ISIR) F-75005 Paris, France <u>nicolasbredeche@upmc.fr</u> stephane.doncieux@upmc.fr

Inria, CNRS, Univ. de Lorraine, Loria, UMR 7503 F-54000 Nancy. France

jean-baptiste.mouret@inria.fr

#### http://pages.isir.upmc.fr/~bredeche/evorobots tutorial/

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan © 2018 Copyright is held by the owner/author(s). ACM ISBN 978-1-4503-5764-77/18/07. https://doi.org/10.1145/3205651.3207866



1



« Evolutionary Robotics aims to apply evolutionary computation techniques to evolve the overall design or controllers, or both, for real and simulated autonomous robots »

Patricia A. Vargas, Ezequiel A. Di Paolo, Inman Harvey and Phil Husbands, 2014, **The Horizons of Evolutionary Robotics**, MIT Press

# 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 Symbrion EU project

#### Stéphane Doncieux

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

#### · Jean-Baptiste Mouret

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



# Motivations: robotics



- Building robots with embodied intelligence
- · Learning with state-of-the-art black-box optimization tools

Pfeifer, R., & Bongard, J. (2006). How the body shapes the way we think: a new view of intelligence. MIT press.

Stulp, F. and Sigaud, O. (2013). Robot Skill Learning: From Reinforcement Learning to Evolution Strategies. Paladyn Journal of Behavioral Robotics. Vol 4 No 1 Pages 49-61.

# Motivations: biology



- ER as a model:
  - modeling evolutionary dynamics, in particular of groups
  - studying the emergence of features
- ER as a tool: optimization and analysis of computational models

Long, J. (2012). Darwin's Devices: What Evolving Robots Can Teach us about the History of Life and the Future of Technology. Basic Books.

Liénard, J. and Girard, B. (2014). A Biologically Constrained Model of the Whole Basal Ganglia Addressing the Paradoxes of Connections and Selection. Journal of Computational Neuroscience. Vol 36 No 3 Pages 445--468.

# Main features of Evolutionary Robotics

Selective pressure	priority to task resolution	or	task resolution secondary (or absent)
Focus	control	and	morphology
Implementation	simulation	or	real world
Space	centralized	or	distributed
Time	off-line	or	on-line

# **Evolutionary Robotics**



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

# Overview

Selective pressure	priority to task resolution	or	task resolution secondary (or absent)
Focus	control	and	morphology
Implementation	simulation	or	real world
Space	centralized	or	distributed
Time	off-line	or	on-line

**Part I:** Fitness function and influence of selection pressure: What do you need to know about evaluation and selection to make an ER experiment successful ?

# Overview

Selective pressure	priority to task resolution	or	task resolution secondary (or absent)
Focus	control	and	morphology
Implementation	simulation	or	real world
Space	centralized	or	distributed
Time	off-line	or	on-line

#### Part II: Encodings of controllers and morphologies: What can you evolve and how ?

# Overview

Selective pressure	priority to task resolution	or	task resolution secondary (or absent)
Focus	control	and	morphology
Implementation	simulation	or	real world
Space	centralized	or	distributed
Time	off-line	or	on-line

**Part IV:** Embodied evolution and collective robotics systems Evolution without a fitness for the design of distributed robotics systems and for modeling evolution of group dynamics.

# Overview

Selective pressure	priority to task resolution	or	task resolution secondary (or absent)
Focus	control	and	morphology
Implementation	simulation	or	real world
Space	centralized	or	distributed
Time	off-line	or	on-line

Part III: Evolution for physical robots and the reality gap How to make it work on real robots ? Fitness function and influence of selection pressure



# Example 1: obstacle avoidance





https://github.com/doncieux/navigation

# Example 1: obstacle avoidance

- How to deal with it ?
  - Change fitness:  $\frac{1}{nb_{coll}+1} * \bar{v}$
  - Make the robot move by default
  - ...



https://github.com/doncieux/navigation

Problem !



# Example 2: Collect ball experiment

Fitness= nb<sub>ball</sub>



https://github.com/doncieux/collectball



https://github.com/doncieux/collectball

# Problem !



# Example 2: Collect ball experiment

- How to deal with it?
  - Decompose the problem
  - Add fitness terms
  - Enhance exploration



https://github.com/doncieux/collectball

# The challenges of selective pressures

Goal of the evolutionary process:

Can we deal with issues in goal definition

and exploration in a task agnostic manner? minimizing a priori knowledge about how to solve the task



# How is fitness evaluated ?

$$f(g) = F(s_0, s(1), \dots, s(T), e_0, e(1), \dots, e(T))$$

- The fitness depends on the genotype and on the fitness function
- But also on:  $s_0 \quad e \quad G(.) \quad T$  $s_0, s(1), \dots, s(T) \quad e_0, e(1), \dots, e(T)$

# How is fitness evaluated ?



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

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

$$f(g) = F(s_0, s(1), \dots, s(T), e_0, e(1), \dots, e(T))$$

# Beyond black-box optimization

To solve the challenges, the selective process can take into account:  $s_0 \ e \ G(.) \ T$  $s_0, s(1), \ldots, s(T) \ e_0, e(1), \ldots, e(T)$ 

Two challenges, two kinds of solutions:

Goal refiner Process helper

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.

Multi-objectivization: a convenient tool to modify selective pressures



- At the end of the run:
  - · Goal refiners: taken into account
  - · Process helpers: ignored

# Overfitting



# Solution to goal definition issues: add « goal refiners »



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

Typical challenges that can be addressed:

- Overfitting & generalisation
- Reality gap

# Encouraging reactivity

- Encouraging robot controllers to react to sensor stimuli
- · Proposition: maximizing the mutual information between sensors and effectors:

$$I(X,Y) = \int_Y \int_X p(x,y) log\left(\frac{p(x,y)}{p(x)p(y)}\right) dxdy$$

# A goal refiner to overcome overfitting

Fitness:

- 1. distance to the goal
- 2. reactivity

Multi-objective EA: NSGA-II

Neuroevolution (HyperNEAT)



Lehman, J., Risi, S., D'Ambrosio, D., & O Stanley, K. (2013). Encouraging reactivity to create robust machines. Adaptive Behavior, 21(6), 484-500.

# A process helper to deal with premature convergence



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.

# Solution to exploration issues: add « process helpers »



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

# Collect ball experiment

Fitness:

1. nb<sub>ball</sub>

2. f<sub>bd</sub>

Multi-objective EA: NSGA-II

Neuroevolution



https://github.com/doncieux/collectball

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.

# Dealing with goal definition and exploration at once

- Changing views:
  - Exploration as a priority: generate all solutions of interest
  - Performance as a secondary, local pressure



- Main ideas:
  - Process helper: selection mostly driven by behavior novelty
  - Goal refiner: a posteriori selection of the most appropriate solution







# Conclusion on selective pressures

Random generation

Variation

Evaluation

Genotype

Phenotype

Initial condition

Environment

₹

Behavior

Fitness

Selection

Z

- The definition of the fitness is critical
- Beyond black box optimization
- Multi-objective framework convenient: multi-objectivization
- Perspective of QD/Illumination algorithms

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.

dia.org/w



# Encodings

Jean-Baptiste Mouret (Inria)









)verview

encodings for evolutionary robotics

- Question: how do we describe a controller and/or a morphology?
- typically, a controller is a graph (e.g. a neural network)
- morphology can also be described by a graph
- we need to encode structures, not just parameters

#### Two families of encodings (for morphology and controllers):

#### I. Direct encodings:

- genotype = phenotype
- example: mutation changes a connection in a neural network

#### 2. Indirect encodings (developmental encodings):

- Genotype i (developmental rules) phenotype (e.g. controller)
- length(genotype) << length(phenotype)</pre>
- can allow to re-use the same genotype (e.g. repetitions, symmetries, etc.)

#### **Related tutorials:**

- Evolution of Neural Networks (Risto Miikkulainen)
- Generative and Developmental Systems (Kenneth Stanley)
- Representations for Evolutionary Algorithms (Franz Rothlauf)

## Fixed topology

- . Choose the topology of a neural network
- L. Consider the weights as a list of parameters (float)
- Mutation: Gaussian noise on one/several parameters



Typical topologies:

- Feed-forward neural network (with hidden nodes)
- Fully connected neural network (recurrent)

Classic variant: fully connected neural network with leaky integrators neurons (Continuous-Time Recurrent Neural Network — CTRNN)

#### **Pros:** simplicity Cons:

- many parameters (does not scale)
- everything influences everything
- competing convention problem (cross-over)



Nolfi S, Floreano (2000). D. Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines. MIT press; 2000.



#### Direct encoding of neural networks

- I. Add a connection between two randomly chosen neurons.
- 2. Remove a randomly chosen connection.
- 3. Add a neuron by splitting an existing connection in two (the connection weight is kept on the two connections).
- 4. Delete a randomly chosen neuron and all related connections.
- 5. Change random weights using polynomial mutation

#### Crossover is not employed.

**Youret, J.B. and Doncieux, S., 2012.** Encouraging behavioral diversity in volutionary robotics: An empirical study. *Evolutionary computation*, 20(1), pp.91-133.

### Directly encoding morphology

- robot := <vertices><neurons> <actuators>
- **vertex** := <x, y, z>
- bar := <vertex | index, vertex 2 index, relaxed length, stiffness>
- neuron := <threshold, synapse coefficients of connections to all neurons>
- actuator := <bar index, neuron index, bar range>a



Direct encoding / Controllers







Direct encoding / Morphology + controllers



NFAT

- Related tutorials:
  - Evolution of Neural Networks (Risto Miikkulainen)

**Stanley KO, Miikkulainen (2002)** R. Evolving neural networks through ugmenting topologies. *Evolutionary computation*. 2002;10(2):99-127.

Direct encoding / Controllers





**Luke S, Hohn C, Farris J, Jackson G, Hendler J. 1997.** Co-evolving soccer softbot team coordination with genetic programming. *RoboCup*-97: *Robot soccer world cup I.* 1998:398-411.





After Mutation

Genetic programming



Direct encoding / Morpholog

## Direct encoding + selective pressure



Indirect Encodings

generation 0

- Classic approach: encoding that favors modularity
- Alternative: Multi-objective Evolutionary Algo. (NSGA-II)
  - performance & total length of connections
- Structural modularity score
  - Newman (spectral optimisation)

Clune\* J, Mouret\* J-B, Lipson H. 2013. The evolutionary origins of modularity. Proceedings of the Royal Society: B 280: 20122863.



#### modularity modularity

#### Direct encoding / Controllers

# Sims' encoding: directed graphs

Genotype: directed graph.





Sims, Karl. (1994) "Evolving 3D morphology and behavior by competition." Artificial life 1.4 (1994): 353-372.

Indirect encoding / Morpholog

Phenotype: hierarchy of 3D parts



curid=694302

# Cellular encoding



(b) Sequential Division 'S' (c) Parallel Division 'F



(d) Division 'T' (e) Division 'A (f) Division 'A





Sruau F. (1994) Automatic definition of modular neural networks. Adaptive behavior. 994 Sep:3(2):151-83.

(odjabachian J, Meyer JA. (1998) Evolution and development of neural controllers or locomotion, gradient-following, and obstacle-avoidance in artificial insects. IEEE ransactions on neural networks. 1998 Sep;9(5):796-812.



00 

Indirect encoding / Controller

#### L-Systems





**-Iornby GS, Pollack JB. (2002)** Creating high-level components with a generative representation for body-brain evolution. *Artificial life*. 2002;8(3):223-46.

#### Gene Regulatory Networks



3ongard J. (2002) Evolving modular genetic regulatory networks. In Proc of IEEE CEC 2002.

Sussat-Blanc, Sylvain, and Jordan Pollack. (2012) "A cell-based developmental model to generate robot norphologies." Proceedings of the 14th annual conference on Genetic and evolutionary computation. ACM, 2012.

**Vattiussi C, Floreano D. (2007)** Analog genetic encoding for the evolution of circuits and networks. *IEEE Transactions on evolutionary computation*. 2007 Oct;11(5):596-607.



Indirect encoding / Controllers

#### Map-based encoding



- high-level abstraction
- inspired by computational neuroscience *models*



**Mouret, Jean-Baptiste, Stéphane Doncieux, and Benoît Girard.** (2010) "Importing the computational neuroscience toolbox into neuro-evolutionipplication to basal ganglia." *Proc. of GECCO 2010.* 

**Sirard B, Tabareau N, Pham QC, Berthoz A, Slotine JJ (2008).** Where reuroscience and dynamic system theory meet autonomous robotics: a contracting vasal ganglia model for action selection. *Neural Networks.* 2008 May 31;21(4):628-41.

#### Indirect encoding / Controller

#### CPPN as an abstraction of development



**Stanley, Kenneth O. (2007)** "Compositional pattern producing networks: A novel abstraction of development." *Genetic programming and evolvable machines* 8.2 2007): 131-162.



## **HyperNEAT**



#### Related tutorials:

 Generative and Developmental Systems (Kenneth Stanley)

Stanley, Kenneth O., David B. D'Ambrosio, and Jason Gauci. (2009) A hypercube-based encoding for evolving large-scale neural networks." *Artificial life* 5.2 (2009): 185-212.

**Clune J, Stanley KO, Pennock RT, Ofria C. (2011)** On the performance of ndirect encoding across the continuum of regularity. *IEEE Transactions on Evolutionary Computation.* 2011 Jun;15(3):346-67.

Indirect encoding / Controllers

repeat for each node

left-right symmetry

#### CPPN for 3D objects





Clune, Jeff, and Hod Lipson. (2011)"Evolving three-dimensional objects with a generative encoding inspired by developmental biology." Proc. of ECAL. 2011.

# CPPN for moving 3D objects

#### We gave evolution four materials:



Cheney N, MacCurdy R, Clune J, Lipson H. (2013) Unshackling evolution: volving soft robots with multiple materials and a powerful generative encoding. In. Proc of GECCO 2013

Auerbach JE, Bongard JC (2010) Evolving CPPNs to grow three-dimensional shysical structures. In Proc. of GECCO 2010.

Indirect encoding / Morphology

#### Jsing a quality diversity algorithm (MAP-Elites)



19

#### Encodings & selective pressure

#### Interactive evolution (no goal):





#### Is the encoding the main limitation/challenge of ER?

Mouret, J.B. and Doncieux, S., (2012). Encouraging behavioral diversity in evolutionary robotics: An empirical study. Evolutionary computation, 20(1), pp.91-133.

Noolley, B. G., & Stanley, K. O. (2011). On the deleterious effects of a priori objectives on evolution and epresentation. In Proc. of GECCO (pp. 957-964). ACM.

#### Conclusion

- Sorry if we skipped your favorite encodings: too many encodings have been proposed
- All encodings encode biases (intentionally or not)
- Biases
- **Current most influential encodings:**
- NEAT for direct encoding
- HyperNEAT / CPPNs for indirect encoding
- Not a big question in the community right now, esp. compared to previous years (no paper about encodings in the CS track this year!)
- Maybe not as important as we thought (selective pressures, stepping stones)





# Evolution. simulators, and the reality gap

erc

Jean-Baptiste Mouret Inria Nancy-Grand Est





#### No simulator

ANR



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

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

#### No simulator

#### **Evolving morphologies**

... in the real world

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



Brodbeck L, Hauser S, lida F (2015) Morphological Evolution of Physical Robots through Model-Free Phenotype Development. PLoS ONE 10(6): e0128444. https://doi.org/10.1371/ journal.pone.0128444 (creative commons)

#### No simulator

evolving walking	Starting	Time (1 run)	Robot	DOFs	Param.
controllers					
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
Proc		Conc			

#### Pros

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

0 h	snake	12	1152
0 h	hexapod	12	135
2 h	quadruped	9	5
Cons			

- slow (too slow?)
- will not be faster next year

- never 100% real

### Evolving 3D programs for 3D printers



Kuehn, T. and Rieffel, J. (2012) Automatically Designing and Printing Objects with EvoFab 0.2", Proceedings of the 13th International Conference on the Synthesis and Simulation of Living Systems (ALife XIII), pp. 372-378

# 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
- Dart (library): <u>https://github.com/dartsim/dart</u>
- Gazebo (GUI): gazebosim.org
- V-Rep (GUI): www.coppeliarobotics.com
- ...

**S. Ivaldi et al. (2014).** Tools for dynamics simulation of robots: a survey based on user feedback. Proc. of Humanoids

J.-.B. Mouret and K. Chatzilygeroudis (2017). 20 Years of Reality Gap: a few Thoughts about Simulators in Evolutionary Robotics. GECCO workshop (SimER) — 2017





7

8

#### The reality gap

... or what always happens with simulators and robots





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.

# Accelerating time

the ideal process

I. Develop / Evolve in simulation

- $\ensuremath{\text{2.Transfer}}$  the result to the reality
  - upload the controller to the robot
- 3. Enjoy!









#### 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.

10

#### But they can agree (sometimes)

#### The reality gap



### 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 have a low probability of working similarly in reality

One of the main problems of ER

#### What can we do?

no simulator better simulator avoid non-transferable solutions robust controllers

#### What can we do?

- no simulator
- simulation then learning on the physical robot
- better simulator
- avoid "bad" solutions
- robust controllers

# What did we try in evolutionary robotics?



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





12

| ∡

# 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. (1995)"Evolving mobile robots in simulated and real environments." Artificial life 2.4: 417-434. Moeckel et al. (2013) "Gait optimization for roombots modular robots—Matching simulation and reality." Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on (IROS), 2013.

Zagal, J. C., and J. Ruiz-Del-Solar (2007) "Combining simulation and reality in evolutionary robotics." Journal of Intelligent and Robotic Systems 50.1.

## Improving simulators

#### The EEA algorithm: active learning of a self-model



Bongard, Zykov and Lipson (2006). Science.

Koos, S., Mouret, JB and Doncieux, S. (2009) "Automatic system identification based on coevolution of models and tests." *Proc. of IEEE CEC.* 

# 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, J. C., and J. Ruiz-Del-Solar (2007) "Combining simulation and reality in evolutionary robotics." Journal of Intelligent and Robotic Systems 50.1.

#### Learning the simulator

= learning the dynamical model of the robot



- I. try the best policy according to the model
  mer new data
- 2. find a policy that maximises the fitness according the simulator

Chatzilygeroudis K, Rama R, Kaushik R, Goepp D, Vassiliades V, Mouret JB. (2017) Black-Box Data-efficient Policy Search for Robotics. arXiv preprint arXiv:1703.07261.

### Improving simulators

#### Pros

#### Cons

mix simulation and reality:<br/>the best of both worlds?the simulator will never be<br/>perfectfaster than learning without<br/>a simulatorif the correction cannot be<br/>applied? (e.g. aerodynamics)

morphological / env. changes learning a simulator is hard!

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

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

# Avoiding bad simulations

I FET SPEE

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

- weep evolution from exploiting simulation artefacts
- 🗯 goal refiner

#### Examples:

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

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

#### Avoiding bad simulations

envelope of noise & minimal simulations

#### Pros

Lightweight simulations

Noise increases robustness and generalization

#### Cons

Hard to set-up

What noise? what is important?

No surprising dynamic effect

Noise makes evolution harder

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

15

# Avoiding bad simulations

the transferability approach

# Maximize fitness Maximize fitness transferability Control approach - 1 objective: covered distance Transferability approach - 2 objectives: covered distance + transferability Image: Control approach - 1 objective: covered distance Image: Covered distance + transferability Image: Image: Covered distance + transf

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





Cully, Clune, Tarapore & Mouret (2015). Nature.

### 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.)

Cully, A., Clune, J., Tarapore, D., & Mouret, J. B. (2015). Robots that can adapt like animals. *Nature*, 521 (7553), 503-507.

# Improving robustness

evolve controllers with online learning abilities

#### **Example:** neural networks with "adaptives synapses"





Floreano, D., & Urzelai, J. (2000). Evolutionary robots with on-line self-organization and behavioral fitness. Neural Networks, 13(4-5), 431–43.

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

#### 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





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

#### Conclusion



Special issue in Artificial Life Journal: Evolution in Physical Systems, 2017 ! Eds. Rieffel, Mouret, Bredeche, Haasdijk



the reality gap

- No perfect approach to cross the reality gap
- Avoiding simulation is materially challenging and slow
- No perfect simulation
- Simulators should give their confidence (and not only a prediction of the fitness)
- it depends on the scientific question!
  - show the potential of a new encoding? a new selective pressure? simulation might be enough
  - solve challenging robotics problem? this needs to work on real robots

Special issue in Artificial Life Journal: Evolution in Physical Systems, 2017 !

27

#### Evolutionary robotics and collective adaptive systems

Tutorial « Evolutionary Robotics », part: "multi-robots" July 2017, Berlin

#### Nicolas Bredeche

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



http://pages.isir.upmc.fr/~bredeche/evorobots\_tutorial/



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.

Definitions



Positive and negative feedbacks

.





SSR/Harvard



LIS/EF

**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.







Attraction

Orientation

Repulsion

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

-			
٦			
,			

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

nicolas.bredeche@upmc.fr

Stigmergy : indirect communication through the environment

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





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

nicolas.bredeche@upmc.fr





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

#### • 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 algorithms for robotics
    - 1. Optimisation viewpoint: meta-heuristic for policy search
    - 2. Modelling viewpoint: theoretical model + simulate behaviours

Scope of this talk

#### 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]



nicolas.bredeche@upmc.fr

#### Off-line evolutionary robotics (the classic approach)

[Nolfi, Floreano 2000][Doncieux et al. 2015]



- 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

nicolas.bredeche@upmc.fr

- Scope [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 for robotics
  - Robustness through redundancy
  - Parallelising actions wrt a task
  - Parallelising learning/optimisation (if any)
- Interests for understanding natural systems
  - ▶ A modelling method for evolutionary adaptation, social learning
  - ► Emphasise the mechanistic aspects (e.g.: coordination behaviours, physical interactions)

icolas.bredeche@upmc.fr

#### Evolutionary robotics for collective systems :

# A dual method

#### Team composition and levels of selection



nicolas.bredeche@isir.upmc.fr

Level of Selection Team Select best teams Select best teams

Waibel et al. (2009



- Optimised and used in a distributed fashion
- A population of robots improving over time wrt. a task to achieve
- > Continuous adaptation to open, possibly changing, environments

nicolas.bredeche@upmc.fr



#### "Vanilla" embodied evolution algorithm

example with a foraging task



The mEDEA algorithm : environment-driven EE

#### What if... we remove the fitness function



Task-driven embodied evolution A vanilla algorithm



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

nicolas.bredeche@upmc.fr

E.g.: #energy items foraged

Bredeche, Montanier (PPSN 2010



#### nicolas.bredeche@upmc.fr





Mathematical and Computer Modelling of Dynamical Systems, Volume 18, Issue 1, 2012

nicolas.bredeche@upmc.fr



# A dual motivation

#### nicolas.bredeche@upmc.fr



nicolas.bredeche@upmc.fr



#### • Lessons learned

- Relevance as an optimisation method (w.r.t. decentralized RL)
- Relevance as an on-line distributed learning method
- About team composition (how to apply selection, how to ensure genetic polymorphism)
- About mechanistic aspects (i.e. cooperation first requires coordination)

#### • Open issues

- Social intelligence (division of labour)
- Behavioural complexity (so far: limited decision-making capabilities)
- Levels of adaptation (lifetime learning vs. evolutionary learning
- Hardware issues (incl. fields of applications, e.g. smart materials)



2



nicolas.bredeche@upmc.fr

« 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, <u>we have to look at artificial ones</u>. » NATURE - VOL 355 - 27 FEBRUARY 1992

#### **Byte-sized evolution**

John Maynard Smith

- Relevance as a modelling and simulation method
  - vs. mathematical modelling
    - simulates mechanistic aspects
  - vs. in vitro or in vivo studies
    - simulates longer evolutionary timescale

nicolas.bredeche@upmc.fr

- Take-home message
  - > Context: collective adaptive systems in open environment
  - Contributions:
    - ER as a modelling tool for understanding natural systems extending classical models with the simulation of behavioura interactions
    - ER as a design tool for making artificial systems, providing distributed online learning algorithms for swarm robotics
- Suggested reading about current trends in ER:

frontiers in ROBOTICS AND AI	REVIEW ARTICLE published: 03 March 2015 doi: 10.3389/frobt.2015.00004
Evolutionary robotics: what, why,	and where to
Stephane Doncieux <sup>1,2</sup> *, Nicolas Bredeche <sup>1,2</sup> , Jean-Baptiste M	ouret <sup>1,2</sup> and Agoston E. (Gusz) Eiben <sup>3</sup>

nicolas.bredeche@upmc.f

# Evolutionary Robotics tutorial Conclusion

N. Bredeche, S. Doncieux, J.-B. Mouret



# Wrapping up

Conclusions and open issues

# Some software tools

- SFERES2: 
  <u>https://github.com/sferes2</u>
  - Software framework in modern C++
  - · As fast as specific code
  - Modules available to evolve robots, examples:
    - Neural network module: 
      <u>https://github.com/sferes2/nn2</u>
    - Simple simulation of a 2-wheeled robot: 
      <u>https://github.com/sferes2/fastsim</u>
  - Code of many experiments on <a href="http://pages.isir.upmc.fr/evorob\_db">http://pages.isir.upmc.fr/evorob\_db</a>
  - Basic experiments to starting playing with ER:
    - Two-wheeled robot maze navigation & obstacle avoidance: 
      <u>https://github.com/</u>
      <u>doncieux/navigation</u>
    - Collect ball experiment: 
      <u>https://github.com/doncieux/collectball</u>
- NEAT & HyperNEAT packages: <a href="http://eplex.cs.ucf.edu/neat\_software/">http://eplex.cs.ucf.edu/neat\_software/</a>

# Recent reviews & introduction papers

- Doncieux S, Bredeche N, Mouret J-B & Eiben AE (2015).
  Evolutionary robotics : what, why, and where to. Front. Robot. Al 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
- Nelson, A. L., Barlow, G. J., & Doitsidis, L. (2009). Fitness functions in evolutionary robotics: A survey and analysis. Robotics and Autonomous Systems, 57(4), 345-370.
- Floreano, D., Dürr, P., & Mattiussi, C. (2008). Neuroevolution : from architectures to learning. Evolutionary Intelligence, 1(1), 47–62. Evolutionary Intelligence, 1(1), 47–62.



# Landmark papers

- 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.
- Lipson, H., and J B. Pollack. (2000) Automatic design and manufacture of robotic lifeforms. Nature 406.6799 : 974-978.
- 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.
- 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.
- Bongard, J., V. Zykov, and H. Lipson (2006). Resilient machines through continuous self-modeling. Science 314.5802 :1118-1121.
- Lehman, J., and Kenneth O. Stanley (2011). Abandoning objectives : Evolution through the search for novelty alone. Evolutionary computation 19.2 (2011) : 189-223.
- Cully, A., Clune, J., Tarapore, D., & Mouret, J. B. (2015). Robots that can adapt like animals. Nature, 521(7553), 503-507.



# Evolutionary Robotics Community

- · Dedicated conferences/tracks:
  - Complex Systems track in ACM Genetic and Evolutionary Computation Conference (GECCO)
  - Evolutionary robotics track at IEEE-WCCI (World Congress on Computational Intelligence)/IEEE-CEC (Congress on Evolutionary Computation)
  - EvoROBOT track in EvoSTAR
- Dedicated journals:
- · Frontiers in Robotics and AI, Evolutionary Robotics specialty section
- Evolutionary Intelligence, Springer
- Mailing lists:
  - General: evoderob@listes.upmc.fr
  - On NEAT: <u>neat@yahoogroups.com</u>

Mouret, J.-B. and Doncieux, S. (2010). SFERESv2: Evolvin' in the Multi-Core World. WCCI 2010 IEEE World Congress on Computational Intelligence, Congress on Evolutionary Computation (CEC). Pages 4079--4086.

# Acknowledgement







Jean-Baptiste Mouret



