Robotique Evolutionniste

Stéphane Doncieux
Plan du cours

1. Algorithmes évolutionnistes
2. Représentation
3. Pressions de sélection
4. De la recherche convergente à la recherche divergente
5. Perspective Robotique évolutionniste & robotique développementale
Robotique
Evolutionniste
Introduction
Evolutionary Robotics aims to apply evolutionary computation techniques to evolve the overall design or controllers, or both, for real and simulated autonomous robots.

Motivations: evolution as a model

- **Biology:** Evolution resulted in agents exhibiting embodied intelligence.

Ernst Haeckel’s Tree of Life
Evolution as a model: Human creativity

Human creativity as the outcome of variation and selection system [Dietrich and Haider 2014]

Evolution in the brain?
- Synaptic selection [Changeux et al. 1973]
- Neural darwinism [Edelman 1987]


Evolution as a model: Development

Motivations: evolution as a model

- **Neuroscience/psychology:**
  - Darwinian principles may be involved in cognitive functions (creativity)
  - Development may rely on variation and selection principles
Motivation: Engineering Learning

• Reinforcement Learning problem:

  Find the policy $\pi^*$ maximizing the cumulative discounted reward over time

• Example: Q-Learning

  $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha (r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t))$

• Notably hard to apply in robotics [Kober et al. 2013]
Motivation: Engineering Learning

<table>
<thead>
<tr>
<th>Trajectory?</th>
<th>states/actions/rewards (per time step)</th>
<th>rewards only (per time step) (return)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perturbation?</td>
<td>action (per time step)</td>
<td>parameter (per time step) (constant)</td>
</tr>
<tr>
<td>Actor-Critic?</td>
<td>act.-critic</td>
<td>direct policy search</td>
</tr>
</tbody>
</table>

- vanilla gradient
- natural gradient
- reward-weighted averaging

- REINFORCE
- eNAC
- GPIC
- PoWER
- PI^2
- CMA-ES
- CEM

Motivation: Modeling
Tuning and analyzing models

- Development of a basal ganglia model compatible with electrophysiological recordings and anatomical data

- Proposed approach:
  1. Mean field model of the whole basal ganglia
  2. Identification of fixed and optimized parameters.

Objectives:
  1. Plausibility of the parameters wrt literature
  2. Plausibility of the behavior wrt literature

Motivations: evolution as a tool

- **Engineering:** efficient Black-box optimization tools …
  - good to tune and learn models
- **Biology/neuroscience/psychology/…:**
  - tuning complex models
  - analyzing models
Evolutionary Robotics

main principles

Random generation

Variation

Evaluation

Selection

Termination

Evaluation

Initial conditions

Environment

Behavior

Robotique Evolutionniste
Algorithmes Evolutionnistes
Features of Evolutionary Algorithms

- Population-based
- Black-box optimization
- Robust to noise and local optima
- Versatile w.r.t. what is optimized

Random generation

Variation

Evaluation

Selection

Termination

Initial conditions

Evaluation

Genotype

Phenotype

Behavior

Environment

Fitness
Some history

• Variation and selection identified as the main driver of species evolution (Darwin 1859)

• Beginning of the XXth century: evolution as an optimization process (Wright 1932, Box 1957)

• Development of the first algorithms:
  • Genetic Algorithms (Univ Michigan, J. Holland 1962)
  • Evolution Strategies (TU Berlin, I. Rechenberg 1965)
  • Evolutionary Programming (Univ. California, L. Fogel 1966)

• Unification of the field: Evolutionary Computation 1993
Evolutionary Algorithm

EvolutionaryAlgorithm()
1. t ← 0
2. RandomInitialization(Pᵣ)
3. for i ← 0 to N
4.   do
5.     Evaluate(Pᵣ(i))
6. while condition not met
7.   do
8.     Q ← ∅
9.     for i ← 0 to N
10.    do
11.       x ← Recombine(SELECT_REPRO(Pᵣ))
12.       x ← Mutate(x)
13.       Evaluate(x)
14.       Q ← x ∪ Q
15.     Pᵣ₊₁ ← SELECT_REPLACE(Pᵣ, Q)
16.   t ← t + 1

Alternatives:
- Random
- Fitness based
- Rank based

Alternatives:
- Elitist on:
  - Q only
  - P and Q
  - Stochastic

ISBN : 978-3-540-40184-1. Springer
How does it work?

Parameter space
How does it work?

Fitness

Parameter space
How does it work?
How does it work?
How does it work?
How does it work?

Parameter space

Fitness
Algorithms

1. Simple genetic algorithm: microbial GA
2. Efficient parameter optimization: CMA-ES
3. Multi-objective optimization: NSGA-II
Algorithms

1. Simple genetic algorithm: microbial GA
2. Efficient parameter optimization: CMA-ES
3. Multi-objective optimization: NSGA-II
Minimalist Genetic Algorithm: Microbial GA

<table>
<thead>
<tr>
<th>Population</th>
</tr>
</thead>
<tbody>
<tr>
<td>01001110</td>
</tr>
<tr>
<td>11100001</td>
</tr>
<tr>
<td>00110011</td>
</tr>
<tr>
<td>01011011</td>
</tr>
<tr>
<td>11011011</td>
</tr>
</tbody>
</table>
Minimalist Genetic Algorithm: Microbial GA

Population

01001110
11100001
00110011
01011011
11011011

1. Random selection

01001110
00110011

Minimalist Genetic Algorithm: Microbial GA

Population

01001110
11100001
00110011
01011011
11011011

1. Random selection

01001110
00110011

2. Ranking

01001110
00110011

Minimalist Genetic Algorithm: Microbial GA

Population

01001110
11100001
00110011
01011011
11011011

1. Random selection

01001110
00110011

2. Ranking

01001110
00110011

Winner

Loser

Minimalist Genetic Algorithm: Microbial GA

Population

01001110
11100001
00110011
01011011
11011011

1. Random selection

01001110
00110011

2. Ranking

01001110
00110011

Winner
Loser

3. Recombination

01001110
00111111

Winner
Loser

Minimalist Genetic Algorithm: Microbial GA

Population
01001110
11100001
00110011
01011011
11011011

1. Random selection
01001110
00110011

2. Ranking
01001110
00110011
01001110
00110011

3. Recombination
01001110
00111111

4. Mutation
01001110
01111111

Minimalist Genetic Algorithm: Microbial GA

Population

01001110
11100001
01111111
01011011
11011011

1. Random selection

01001110
00110011

2. Ranking

Winner
Loser

01001110
00110011

3. Recombination

Winner
Loser

01001110
00110011

4. Mutation

01001110
01111111

Winner
Loser

01111111
00111111

Algorithms

1. Simple genetic algorithm: microbial GA

2. Efficient parameter optimization: CMA-ES

3. Multi-objective optimization: NSGA-II
CMA-ES

\[ x_k^{(g+1)} = \langle x \rangle_w^{(g)} + \sigma^{(g)} B^{(g)} D^{(g)} z_k^{(g+1)} \]
\[ \sim \mathcal{N}(0, C^{(g)}) \]

\[ p_c^{(g+1)} = (1 - c_c) \cdot p_c^{(g)} + c_c \cdot \frac{c_w B^{(g)} D^{(g)} \langle z \rangle_w^{(g+1)}}{\frac{c_w}{\sigma^{(g)}} \left( \langle x \rangle_w^{(g+1)} - \langle x \rangle_w^{(g)} \right)} \]

\[ C^{(g+1)} = (1 - c_{cov}) \cdot C^{(g)} + c_{cov} \cdot p_c^{(g+1)} \left( p_c^{(g+1)} \right)^T \]

\[ p_{\sigma}^{(g+1)} = (1 - c_\sigma) \cdot p_{\sigma}^{(g)} + c_\sigma \cdot \frac{c_w B^{(g)} \langle z \rangle_w^{(g+1)}}{\frac{c_w}{\sigma^{(g)}} \left( \langle x \rangle_w^{(g+1)} - \langle x \rangle_w^{(g)} \right)} \]

\[ \sigma^{(g+1)} = \sigma^{(g)} \cdot \exp \left( \frac{1}{d_\sigma} \cdot \frac{\| p_{\sigma}^{(g+1)} \|}{\hat{X}_n} - \hat{X}_n \right) \]

CMA-ES

Algorithms

1. Simple genetic algorithm: microbial GA

2. Efficient parameter optimization: CMA-ES

3. Multi-objective optimization: NSGA-II
Multi-objective optimization

Dominance relation

Multi-objective optimization

Dominance relation

Multi-objective optimization

Dominance relation

Solutions dominating a

Solutions neither dominated nor dominating a

Solutions dominated by a

Solutions neither dominated nor dominating a

NSGA-2

Robotique Evolutionniste
Représentation
Alternatives

- Bit string (Genetic algorithms)
- Vector of real parameters (Evolution Strategies)
- Graphs (Evolutionary Programming, Neuroevolution)
- Programs: trees, linear programs (Genetic Programming)
- Your new great representation …
Neuroevolution

\[ p_i = \sum_j w_{ji} o_j \]

\[ o_i = \frac{1}{1 + \exp(-p_i - \lambda \beta_i)} \]
Neuroevolution: NEAT

• Main features:
  • Historical marking:
  • Cross-over
  • Speciation
  • Starting simple

Neuroevolution: NEAT


Neuroevolution: NEAT

Neuroevolution

Neuron

\[ \sum \]

\[ W_1 \]
\[ W_2 \]
\[ W_{n-1} \]
\[ W_n \]

Mutations

Crossover

Neural network

HyperNEAT

1) Query each potential connection on substrate

2) Feed each coordinate pair into CPPN

3) Set weight of connection between \((x_1, y_1)\) and \((x_2, y_2)\) to value of output

Examples of pictures generated with CPPNs

http://picbreeder.org

In: Evolutionary Computation journal, MIT Press, 2011
To go further
Robotique Evolutionniste
Pressions de sélection
Collect ball experiment

Fitness = $n_{ball}$

https://github.com/doncieux/collectball
The fitness function has two roles:

- It defines the goal: ✓
- It drives the search: ✗
The fitness function has two roles:
• It defines the goal ✓
• It drives the search ✗
Solution:
add « process helpers »

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

Solution:
add « process helpers »

Behavioral diversity:

\[ f_{bd}(i) = \frac{1}{N} \sum_{j \in Pop} d(beh_i, beh_j) \]


Collect ball experiment

Fitness:
1. $n_{ball}$
2. $f_{bd}$

Multi-objective EA:
NSGA-II

Neuroevolution

https://github.com/doncieux/collectball
Process helpers

• **Task specific:**
  • Fitness shaping [Dorigo and Colombetti, 1994] [Nolfi 1997]
  • Incremental evolution [Harvey et al. 1994]
  • Incremental MOEA [Barlow et al. 2004]
  • Staged MOEA [Mouret et al. 2006]

• **Task agnostic:**
  • Competitive coevolution [Cliff and Miller 1996]
  • Behavioral diversity [Mouret and Doncieux, 2009]
  • Novelty objective [Mouret 2011, Lehman et al. 2013]
  • Coevolution environment/controller [Berlanga et al. 2000]

Another problem
Another problem
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

A goal refiner to overcome overfitting

A goal refiner to overcome overfitting
Goal refiners

- **Task specific:**
  - Behavioral consistency [Ollion et al. 2012]
  - Noise and occlusion [Ollion 2013]

- **Task agnostic:**
  - Coevolution model-test [Nardi et Holland 2008, Koos et al. 2009]
  - Envelope of noise [Jakobi 1997]
  - Reactivity [Lehman et al. 2013]
  - …
Conclusion on selective pressures

- The definition of the fitness is critical
- Beyond black box optimization
- Multi-objective framework convenient: multi-objectivization

Robotique Evolutionniste
De la recherche convergente à la recherche divergente
**Picbreeder: A case study in collaborative evolutionary exploration of design space.** Evolutionary Computation, 19(3), 373-403.
PicBreeder

From convergent to divergent search

Looking for the optimal solution

Looking for novel or original solutions


Novelty search

- Objective to maximize: distance towards the k nearest points in population+archive (novelty)

- Archive augmented with individuals having a high novelty

S’adapter à des pannes

Robotique évolutionniste
Application en robotique développementale
Robotique développementale

Construire un programme informatique permettant à un robot d’être « éduqué » comme un enfant.

Développement

Identifying important features & transferring knowledge

DREAM overview

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

Main ideas:
- evolutionary approach to bootstrap cognition
- redescription of acquired knowledge
- alternation between
  - « daytime »: active interaction
  - « nighttime »:
    - analysis of past events
    - knowledge consolidation
    - simulation of new behaviors
DREAM Overview

Collective scale

Daytime experience (large batch)

Consolidated knowledge
- task-relevant features
- task contexts
- abstract knowledge
- new motivations

Behavior exploration
Knowledge improvement
Knowledge adaptation
Knowledge validation

Knowledge sharing between robots:
- better generalization
- faster learning

No initial policy
No single task
Motivations:
- curiosity
- satisfying humans
- global mission

New situation:
- no reprogramming
- fast adaptation

Individual scale

Small batch
Skill

Nighttime

Dream
Knowledge restructuring
Transfer from STM to LTM

Daytime experience (large batch)
DREAM objectives

System 1
« Implicit cognition »

System 2
« Explicit cognition »

Redescription

Sense
- video cam.
- laser
- ...

Plan
1. update world representation
2. plan actions

Act
Send commands to motors

Bootstrap with low-level and agnostic representations and progressively build more adapted representations

Stulp, F, Hospedales, T. « Dual-Process Representational redescriptions », CDS Newsletter, Fall 2015
Doncieux, S. « Representational redescription: the Next Challenge ? Summary and reply », CDS Newsletter, Fall 2015
Generates examples of behaviours
Discrete actions and sensors to consider
Passive analysis
Representation redescription

Learning
Direct policy search (neuroevolution)
Task-agnostic representations
Slow learning
Limited generalization

Passive analysis
Representation redescription

Learning
Discrete reinforcement learning
Task-specific representations
Fast learning
Good generalization

IEEE Transactions on Cognitive and Developmental Systems
Learning to manipulate objects

Learning to manipulate objects
Learning to manipulate objects
Learning to manipulate objects
Babbling
(the first generation)
Software tools

• **SFERES2**: [https://github.com/sferes2](https://github.com/sferes2)
  
  • Software framework in modern C++
  
  • As fast as specific code
  
  • Modules available to evolve robots, examples:
    
    • Neural network module [https://github.com/sferes2/nn2](https://github.com/sferes2/nn2)
    
    • Simple simulation of a 2-wheeled robot [https://github.com/sferes2/fastsim](https://github.com/sferes2/fastsim)
    
    • Code of many experiments on [http://pages.isir.upmc.fr/evorob_db](http://pages.isir.upmc.fr/evorob_db)

Merci pour votre attention !

stephane.doncieux@upmc.fr
https://twitter.com/SDoncieux
http://people.isir.upmc.fr/doncieux