Adaptive Systems

Ezequiel Di Paolo
COGS

Evolutionary Robotics
1. The Basics

Next few lectures…

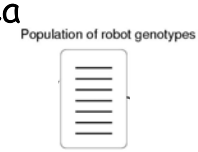
… will be looking at Evolutionary Robotics.

- Motivations. Techniques. "Classical" examples.
- Real robots vs. simulations. Transferring controllers to reality. Minimal simulations.
- Minimal cognitive behaviour.
- Hot topics. GasNets, plastic controllers, body morphology, fast oscillations, spiking neurons, development, adaptation to radical perturbations

Evolutionary Robotics

- The use of evolutionary computing techniques for developing autonomous adaptive robots (control system and properties of sensors, motors, and body).
- **Engineering:** Alternative to traditional design techniques
- **Science:** Less prejudiced way of exploring wide class of mechanisms for generating adaptive behaviour. Synthesize and analyse.

Basic idea



Motivations

- **The Design Problem.** Hand design of systems involving many interactions between many constituent parts is very hard. Probably too hard. Yet systems with these properties appear to be necessary to generate interesting adaptive behaviour in autonomous agents (and possibly have other engineering potential). Some kind of automation is required. Artificial evolution is one way of doing it.

Motivations

- **Fewer preconceptions.** Evolutionary methods allow us to start with as few preconceptions as possible. The designer evaluates appropriate behaviour rather than specifying how it is realised. The result is often surprising precisely because of the preconceptions of the designer which would have gone into the design itself in other methodologies.

Motivations

- **Demonstrations of mechanisms.** Evolutionary robotics can provide new demonstrations of mechanisms underlying adaptive behaviour. Analysis is important. Mechanisms are often likely to be unintuitive from a traditional standpoint. It can also explore what known biological mechanisms might be doing when we close the sensorimotor loop in a whole, situated agent.

Fundamental research issues

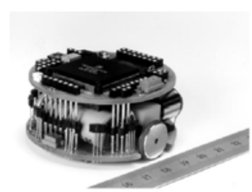
- Genotype to phenotype mappings
- Analysis of evolutionary processes
- Analysis of evolved robots
- Adaptation to changing environments
- Co-evolution of control architectures and body plans
- Interactions of evolution, development, learning

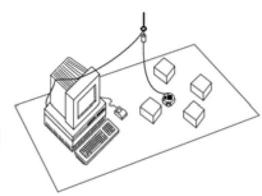
Fundamental research issues

- Evolvable hardware in robotics
- Simulation-reality transfer
- Comparison of evolutionary robotics methodologies
- Scaling to complex behaviours
- Nature of control systems building blocks
- Fitness function design (The art of ...)

Floreano and Mondada

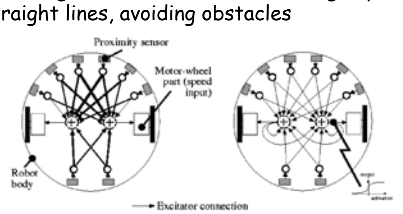
- EPFL, Lausanne, Artificial evolution using a real Khepera robot (1995).





Floreano and Mondada

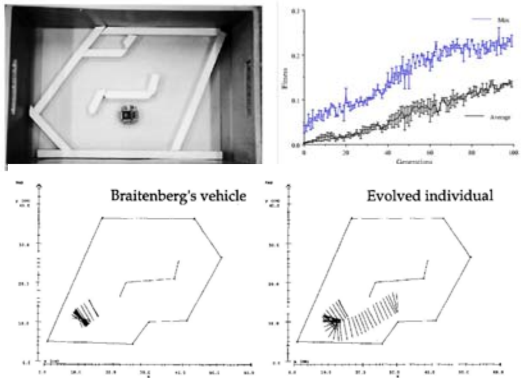
- Fitness = $\Sigma V (1 - \sqrt{\Delta v})(1 - i)$
- V = Avg. rotation speed of wheels, $\Delta v = |v1 - v2|$, i = reading of most active sensor ... High speed, straight lines, avoiding obstacles



Floreano and Mondada

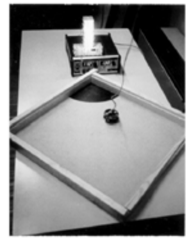
- Evolve weights and thresholds in a fixed architecture network using straightforward encoding
- Standard GA, PopSize 80, 100 generations, 39 mins per generation. Good results after ~50 generations

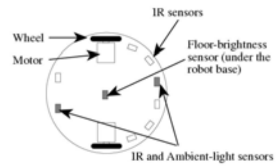
1. Download genotype
2. Run network controller, recording fitness
3. Five seconds of random movement
4. Goto 1.



Battery recharge

- Fitness: = ΣV (1 - i); stop run if battery is flat





Battery recharge

- Simulated battery runs out linearly in 20 secs.
- Robot must step on black area to recharge
- Robot wanders around until level is close to zero (depends on distance) then speeds towards recharging area.

Randall Beer

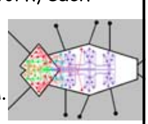
- Case Western Reserve University: Evolution of chemotaxis, legged locomotion, sequential behaviour; mostly in simulations
- Continuous-time recurrent neural networks (CTRNNs, aka. dynamical neural networks) to generate adaptive behaviour:

$$\tau_i \frac{dy_i}{dt} = -y_i + \sum_j w_{ji} z_j + I_i; \quad z_j = \frac{1}{1+\exp[-(y_j + b_j)]}$$

- Use evolution to find weights, thresholds, time constants in fully connected architectures. Typical popsize 500, simple bit encoding, moving to real-valued encoding in later experiments

Simulated "Cockroach"

- Six legs. Legs generate forces which move the body
- Each leg has 3 effectors: up or down, clockwise torque about leg's joint, counter-clockwise torque.
- Each leg controlled by five neuron network, each neuron has input from angle sensor.
- Identical controllers for each leg, symmetrical connections between them.



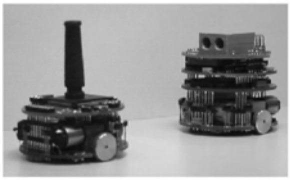
- Maximise the forward distance moved in a fixed time

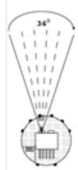
Advice on CTRNNs

- Weight range
  - Neurons are bi-stable with self connections greater than 4. Best choose weights to be in range [-n,n] with n>4.
- Time constants, integration method
  - Allow slow time constants if task requires it. Roughly up to the length of the whole task.
  - Maximum integration step < minimum tau/2
  - Choose linear or exponential scaling
- Sigmoid function
  - Generate a large (1000) array of floats and fill it with the values of the sigmoid function.
  - Use this array. Don't calculate the function every time.
- Centre-crossing seeding.
  - Initial population. Make bias b_i = -∑ wji/2

Coevolution

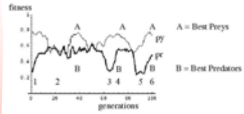
- May lead to "arms races". Often thought of as a possible route towards open-ended complexity. In fact, it may also lead to equilibria or cyclic dynamics.
- Example: pursuit-evasion. Nolfi and Floreano (1998)

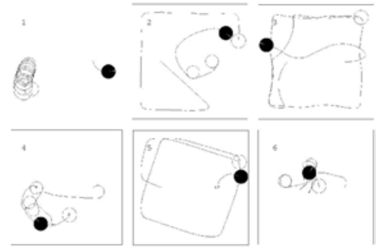




- Fitness is no clear measure of complex behaviour

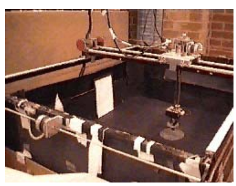
Coevolution





The Sussex approach

- Largest concentration of researchers on Evolutionary Robotics in the galaxy.
- *Centre for Computational Neuroscience and Robotics* (CCNR). Created 1996. Joint venture BIOLS + COGS. Strong link with neuroscientists (T. Collett, M. O'Shea).



Sussex: some topics

- Minimal simulations (Jakobi, Husbands)
- GasNets (Husbands, T. Smith, Phillipides)
- Evolvable Hardware (Thompson)
- Evolutionary dynamics (Harvey, Barnett, T. Smith)
- Insect-like navigation (Dale, L. Smith)
- Theoretical issues (Seth, Harvey, Husbands)
- Social behaviour (Quinn, Di Paolo)
- Learning, adaptation (Bird, Tuci, Di Paolo)
- STDP, spiking neurons (Di Paolo)
- Active vision, arm movement (Buehrmann)
- Powered passive dynamic walkers (Vaughan)

Sussex: some prejudices

- Explore (e.g., unusual bio-inspired mechanisms)
- Dynamical systems approach
- Complex behaviours are not the only interesting result ("scaling-up" often a side issue)
- Minimum designer interference
- Analyse (as much as possible/desirable)
- Avoid trivialized, vacuous connectionism

The use of GAs

- Some work in ER aims at obtaining and studying robots that do something interesting
- Using a GA in this case is only a technique
- Using other search algorithms is equally valid (hybrid algorithms, "netcrawlers", etc.)
- But sometimes the use of an evolutionary algorithm is more significant and adds important value to the result (eg. Looking a problems related to evolutionary biology).

ER and other disciplines

- From the point of view of AI and cognitive science, ER attempts to build increasingly complex autonomous robots.
- From the point of view of other fields, such as neuroscience, ER can be a useful exploratory tool for understanding mechanisms. Eg. How will a network of spiking neurons function in a integrated robot? This can be a useful line of research whilst the behaviours explored remain simple.

Summary

- Human design: Complexity is often handled using the *Divide and Conquer* approach. Sub-problems may be easier to manage but must be relatively independent from each other.
- Cognition seems to demand highly integrated systems, made out of many components and in complex relations with the environment.
- Classical AI and Behaviour-Based Robotics, still using the *Divide and Conquer* approach.

Summary

- ER aims at evolving *whole systems*. Evolution tends to find highly integrated, distributed, non-modular and surprising solutions to the problem of generating adaptive behaviour. Hence, an interesting tool from the point of view of engineering and biology.

Open questions

- Evolved structures are difficult to analyse. Must perform a variety of tests at different levels (behaviour, analytical, etc.) to approach an explanation. Still easier than analysing a real animal but no guarantee that it will stay that way.
- Scaling problem again + time constraints. How far can you go using evolution? Open question.