Non Symbolic AI - Lecture 15

Final Lecture.
Exam/coursework advice + more on Neutral Networks + robot videos

Final seminar session will be a voluntary one this afternoon Monday 23 May at 2pm in PEV1-1A1

"Agony Aunt session": come if you have any specific NSAI questions at all – except that you cannot ask too specifically about the coursework assignment, only more general questions about GAs and ANNs

What should you write in the coursework writeup?

Firstly, your program code should be clear and commented – so the writeup should not just be a further description of the code

Secondly, even if your code does not work as you had hoped, you can still gain marks from a good writeup

Demonstrate your ***understanding!***

We want to see that you understand the ***basics*** of ANNs, what they can be used for.

The ***basics*** of a learning algorithm such as backprop

The ***basics*** of a GA

The ***basics*** of applying an ANN to a Boolean problem such as this.

Use your own words

Straight ***repetition*** from the lecture notes or a textbook is a ***bad idea!***

It suggests that you have memorised the words ***without*** understanding them.

Your ***own*** way of describing things will be much more impressive!

Did you think beyond the problem?

Was the problem a sensible one?

Was backprop a sensible learning algorithm?

Was a GA a sensible idea?

Would you expect all problems to be like this one, or different?

Did you think beyond your results?

Are you happy with your results? Why?

Were you lucky, or unlucky?

If you did it again once, or many times, would you expect the same answers? Why/why not?

The more intelligent comments that you can make, the better.

Don’t leave it to the last minute!

You should aim to basically finish – and ***print*** out your submission – at least 1, pref 2 days before the deadline.

For some amazing reason, computers and printers break down just before a deadline – it is ***your responsibility*** to anticipate this!

Then with good luck, you may think of some improvements in the last 2 days for a better version – but you are not ***relying*** on good luck.

Exam technique -- Revision

The Exam is due Tue 21 June.

Pointers to previous exam pages via the course web page.

90 minutes, choose Two out of Three questions.

Third optional question is an essay question, with several possible titles given.

Advice

Taken from:-

www.informatics.sussex.ac.uk/doc/unseen_exams.php

Revise Non-Sym AI alongside the other courses

A > B > C > NSAI > A > B > C > NSAI > A > B > ….

Revision is done by ***Writing!***

Take your lecture notes, or textbook or any written material you want to revise – and ***write notes*** in your own words to summarise what is said.

Input Information usually doesn't stick in your brain unless you ***output*** something from it – and writing notes is the best way.

Summarise in notes

First time round, just summarise a paper or a lecture into just a page or two.

Then start again: see if you can summarise your summary into less than a page.

Then start again: see if you can summarise your new summary into a small postcard – in all this, the ***summarising*** makes sense of it, the ***writing*** makes it stick!

Practice an Exam

Take an old NSAI exam – lock yourself in a room for 90 minutes – and see how well you do.

Read the questions.

Manage your time.

Don't spend so much time answering your 'favourite question' that you can write only scrappy notes for the other question you choose.

Answering a question

Plan each answer. Jot down a skeleton answer-plan, on a page which you will cross out as rough work. Especially important for an Essay-answer.

Answer the question on the exam paper - not the one you were expecting to find on the paper.

Check how many points are allocated for each part of a multi-part question – and allocate your time accordingly

Use any spare time to re-read your answers, and improve them

Before the Exam

Do your revision in good time – finish a couple of days before the exam

Then come out of your hole, take a break, get some exercise, get a good night's sleep before the exam.

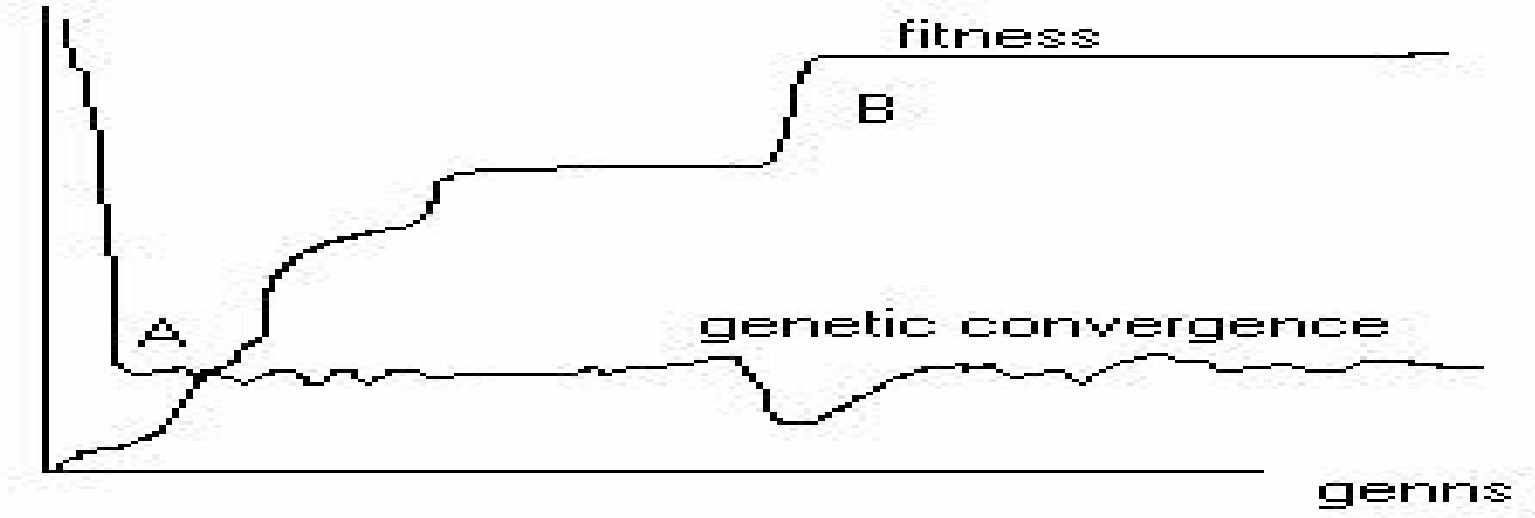
… and ***Don't Worry … !***

Bit more on Neutral Networks

(Followed by some robot videos)

Basics of Neutral Networks covered in last lecture. Here (at high speed) is a reprise of that, plus a bit more.

Genetic Convergence is not Stasis

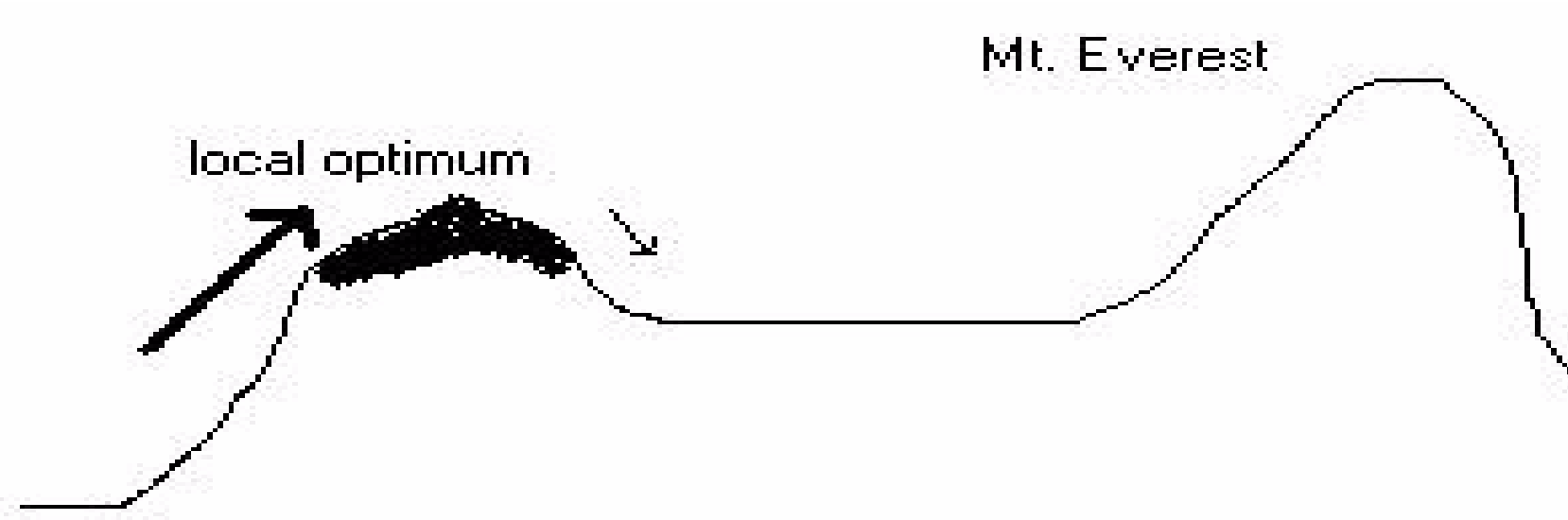


B is the point of convergence (defn 2), often after 'punctuated equilibria'

A is the point of genetic convergence (defn 1), which may well be (surprisingly?) within the first 10 or so generations !

Optimal mutation rates

Continuing SAGA ideas, in a fitness landscape you can have too little mutation (relative to selection:-



… or too much mutation

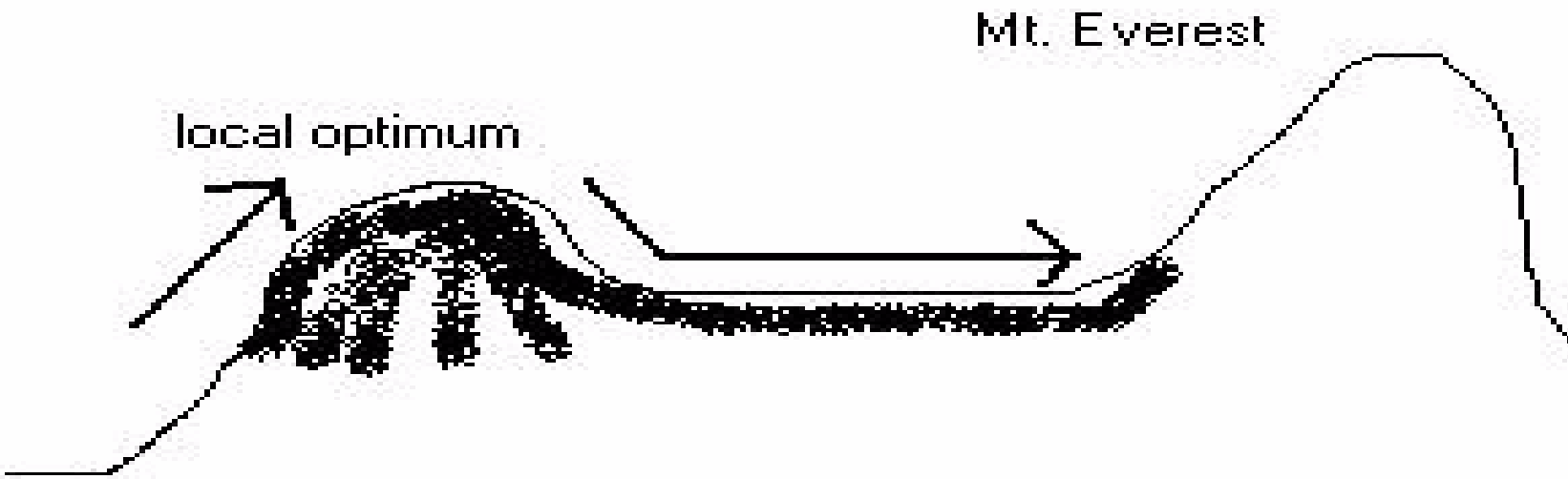
… or you can have too much mutation …

(arrow up the hill is selection, arrow down the hill is mutation)

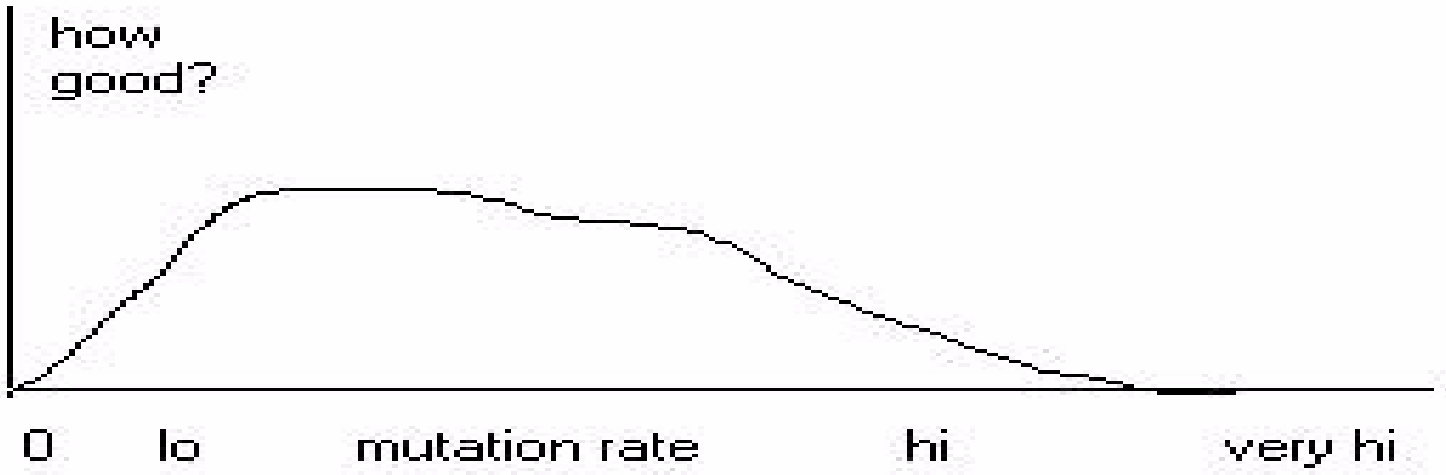
… or just about the right amount …

... or you can gave round about the right amount, to avoid losing height (fitness) gained, but promoting search along **ridges** -- which may lead to higher ground :-

Balance between **exploration** and **exploitation**



Why should there be an optimal mut-rate?



Mutation rate too low, in limit zero, would mean no further change, evolution ceases -> no good

Mutation rate too high, eg every bit flipped at random, implies random search -> no good.

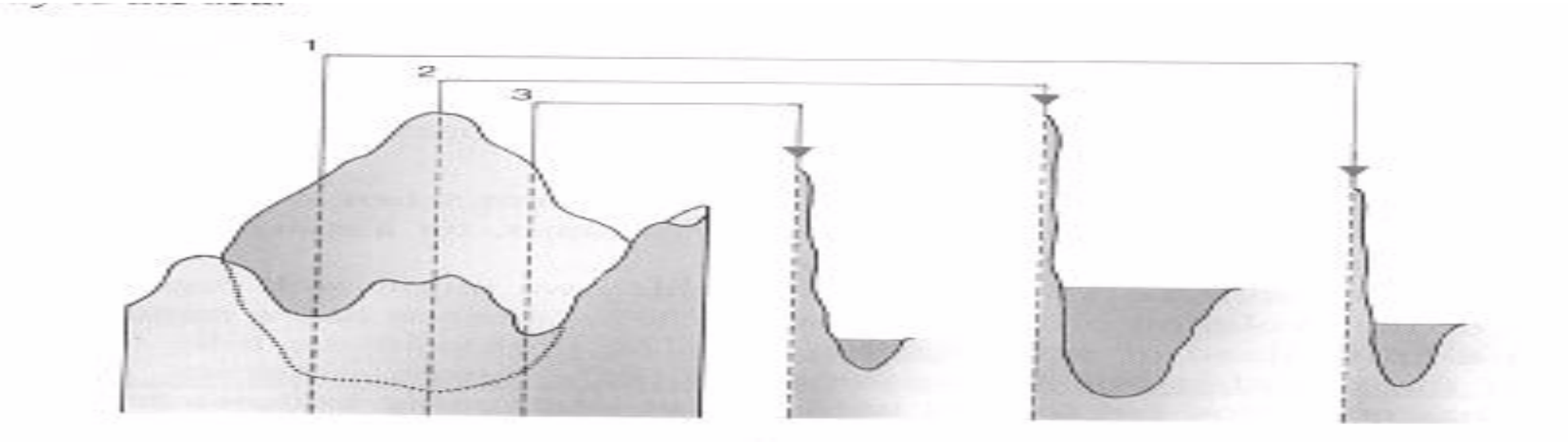
High dimensional landscapes

We can visualise ridges in the 3-D landscapes (Himalayas, South Downs) that the metaphor of fitness landscapes draws upon.

But in 100-D or 1000-D landscapes things can be very significantly different.

In particular you can have ridges in all sorts of directions.

Ridges in high-dimensional landscapes

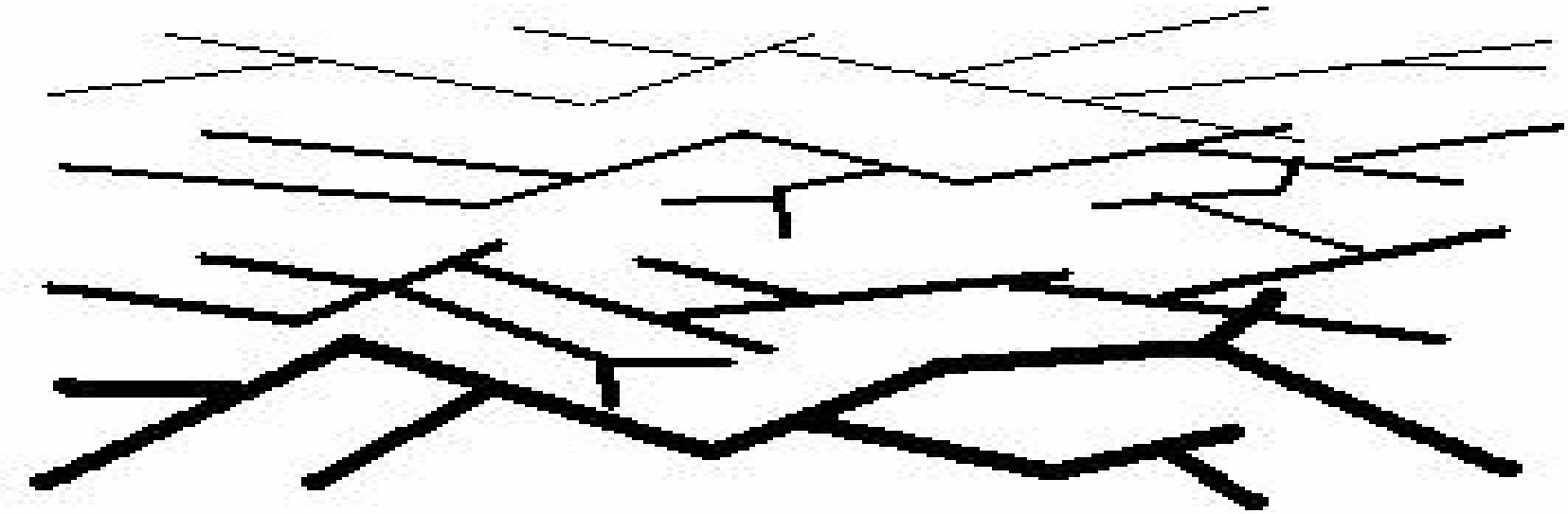


Going from 2-D to 3-D allows extra opportunities for "bypasses around a valley without dropping height"

Going up to 100-D or 1000-D potentially allows many many more such opportunities -- hyper-dimensional bypasses.

(pic borrowed from *Steps Towards Life*, Manfred Eigen Oxford Univ Press 1992)

The New Picture



IF there is lots of neutrality of the right kind, then there are lots of *Neutral Networks*, connected pathways of neutral mutations running through the landscape at one level --

Right kind of redundancy

- Multi-storey building example – some corridors lead to staircases
- When this is so, ‘going along the flat’ buys you something
- Of course, if corridors lead to flat plains without a staircase in sight, you are wasting your time!

The First claim for Neutral Networks

(1) **The Formal claim**

It can be demonstrated indisputably that **IF** a fitness landscape has lots of neutrality of a certain kind, giving rise to Neutral Networks with the property of constant innovation

THEN the dynamics of evolution will be transformed (as compared to landscapes without neutrality) and in particular populations will not get stuck on local optima.

The above would be merely a mathematical curiosity unless you can also accept:-

The Second claim for Neutral Networks

(2) **The Empirical claim**

Many difficult real design problems
(..the more difficult the better...)
in eg evolutionary robotics, evolvable hardware, drug design
--- have fitness landscapes that naturally (ie without any special effort) fit the bill for (1) above.

I make claim (2), but admit it is as yet a dodgy claim!

Recently some supporting evidence.

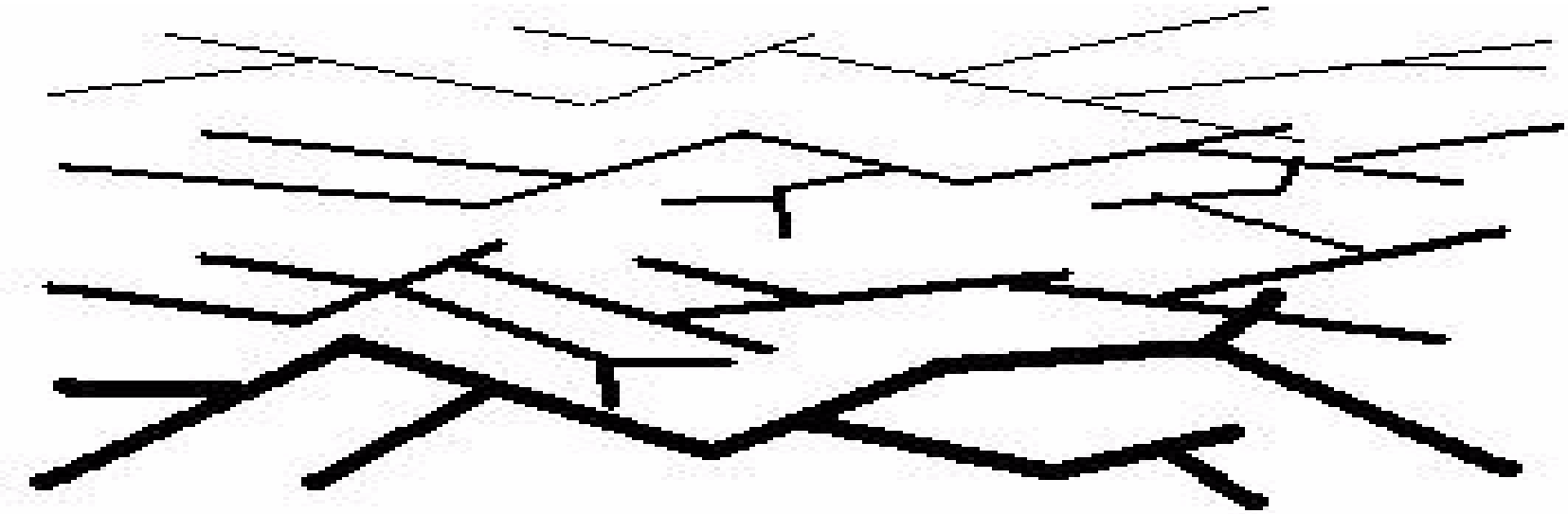
Recent Research on Neutral Networks

One of the first demonstrations of the formal claim was in an EASy MSc dissertation by Lionel Barnett 1997.

See full dissertation, and shorter version for Alife98 conference, on his web pages
http://www.cogs.susx.ac.uk/users/lionelb/

and Neutral Network bibliography via
http://www.cogs.susx.ac.uk/lab/adapt/nn.html

The New Picture



IF there is lots of neutrality of the right kind, then there are lots of *Neutral Networks*, connected pathways of neutral mutations running through the landscape at one level --

… percolation …

-- and *lots and lots* of these NNs, at different levels, **percolating** through the whole of genotype space, passing close to each other in many places.

Without such neutrality, if you are stuck at a local optimum (ie no nbrs higher) then there are only N nbrs to look at
BUT WHEN you have lots of neutrality, then without losing fitness you can move along a NN, with nearly N new nbrs at every step -- 'constant innovation'.

Basically, you never get stuck !

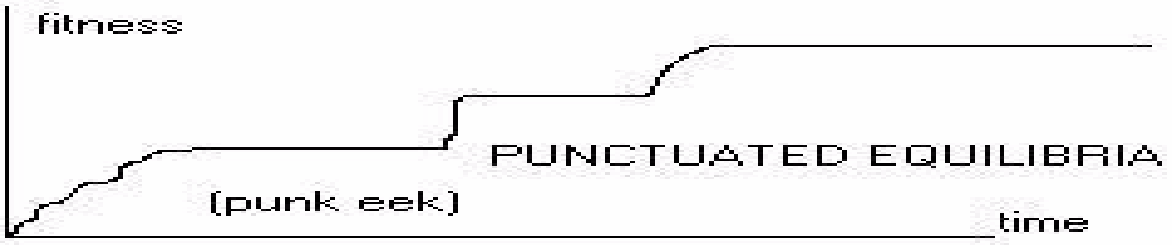
Right kind of redundancy

- Multi-storey building example – some corridors lead to staircases
- When this is so, ‘going along the flat’ buys you something
- Of course, if corridors lead to flat plains without a staircase in sight, you are wasting your time!

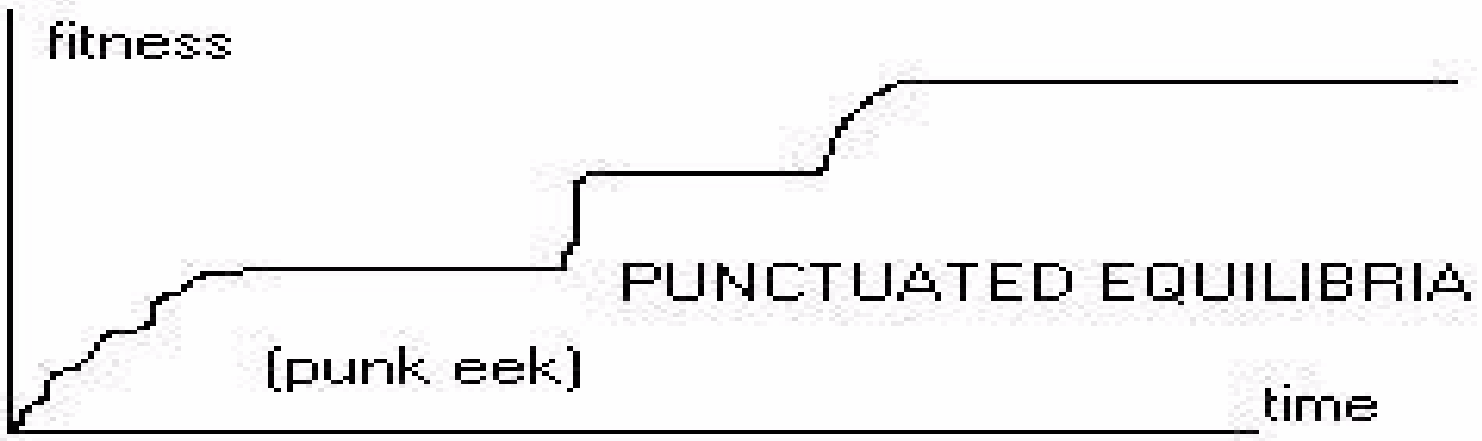
What happens?

Roughly speaking, in such a landscape the population will quickly 'climb onto' a ridge slightly higher than average, then move around neutrally 'looking for a higher nbr to jump to'.

You might have to wait a while (even a long while...) but you will not get stuck for ever. When eventually one of the popn finds a higher NN, the popn as a whole 'hops up‘ and carries on searching as before



Punk Eek



...and significantly, in many real GA problems this is just the sort of pattern that you see.

The horizontal bits are not (as many thought) just standing still waiting for luck --- rather 'running along NNs waiting for luck'

Ruggedness versus Neutrality

Lionel Barnett's NKp landscape gives an abstract framework in which one can tune independently:
K for ruggedness and p for degree of Neutrality.

There are various standard measures for ruggedness
e.g. *autocorrelation* -- roughly, a measure of how closely related in height are points 1 apart, 2 apart, ...10 apart...

Amazingly, for fixed N and K, when you tune parameter p all the way from zero neutrality up to maximum neutrality the autocorrelation remains (virtually) unchanged.

Same ruggedness but different dynamics

Yet as you change the neutrality p, despite having the same ruggedness the *evolutionary dynamics changes completely* -- for zero neutrality the population gets easily stuck on local optima, for high neutrality it does not.

Clearly neutrality makes a *big difference* -- yet this has been completely unknown to the GA community, who have only worried about ruggedness.

Indeed all the typical benchmark problems used to compare different GAs have no neutrality at all.

Net-crawlers

- The EH example was basically a bastardised version of a GA – a **net-crawler** – equivalent to a Steady-state GA with population size 2.

- Lionel Barnett, in his thesis, showed that for a particular class of abstract fitness landscape (epsilon-correlated) that had many NNs, s.t. that the population could jump from one to the next, then **provably** the best search method was a net-crawler with a specific rate of mutation

Net-crawling mutation rate

- Optimal rate is **provably** (under certain assumptions):-
- Mutate exactly n (an integer) bits on genotype
- … where n is chosen so as to make the percentage of neutral mutations as close as possible to 37% (1/e)
- (using plausible assumptions) this can be calculated on the fly, keeping track of how many recent mutations were neutral and adjusting mutation up/down accordingly
- For the EH example, this looks like suggesting 3 mutations !!!

Summary on NNs

Please distinguish between

1. The **FORMAL claim**, proven without doubt: that fitness landscapes full of neutral networks of the right kind completely alter evolutionary dynamics
2. The **EMPIRICAL claim,** that many real-world difficult design problems, with (near-)binary encodings, do in fact have lots of neutral networks of the right kind.

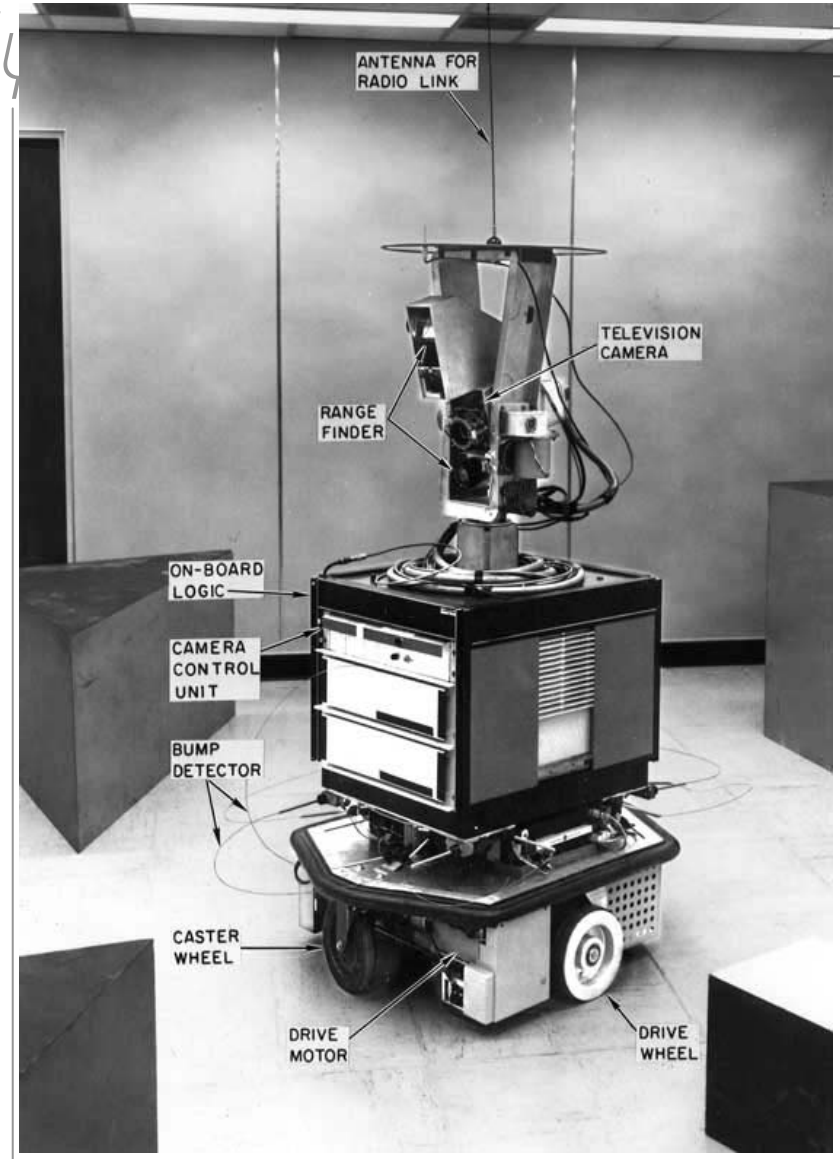
Classical AI

When building robots, the Classical AI approach has the robo
as a scientist-spectator, seeking information from outside.

"SMPA" -- so-called by Brooks (1999)

- S sense
- M model
- P plan
- A action

Shakey

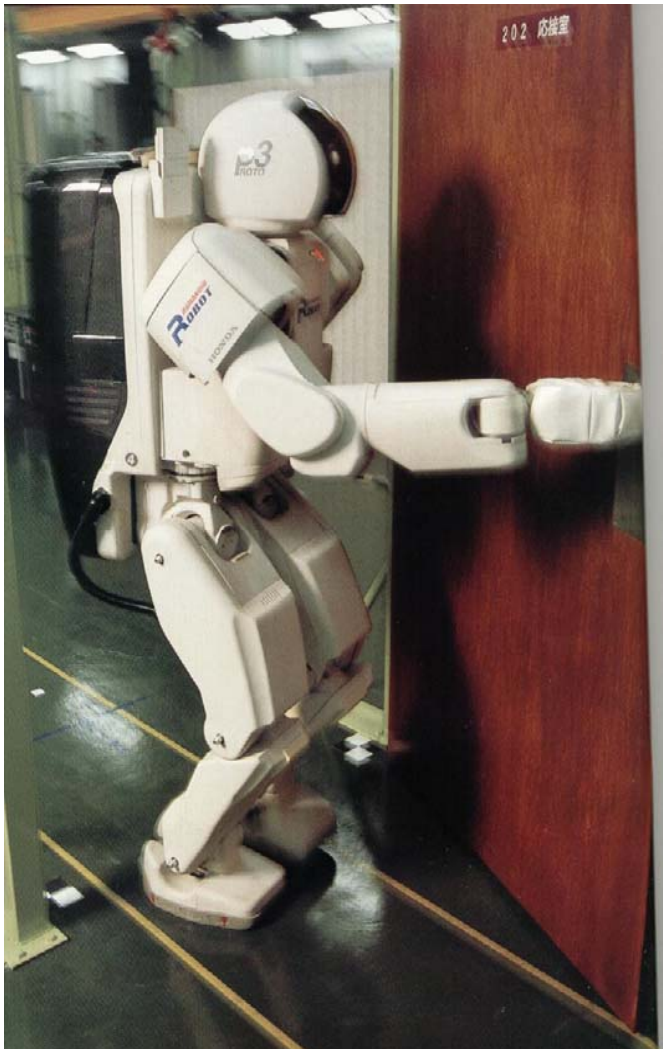


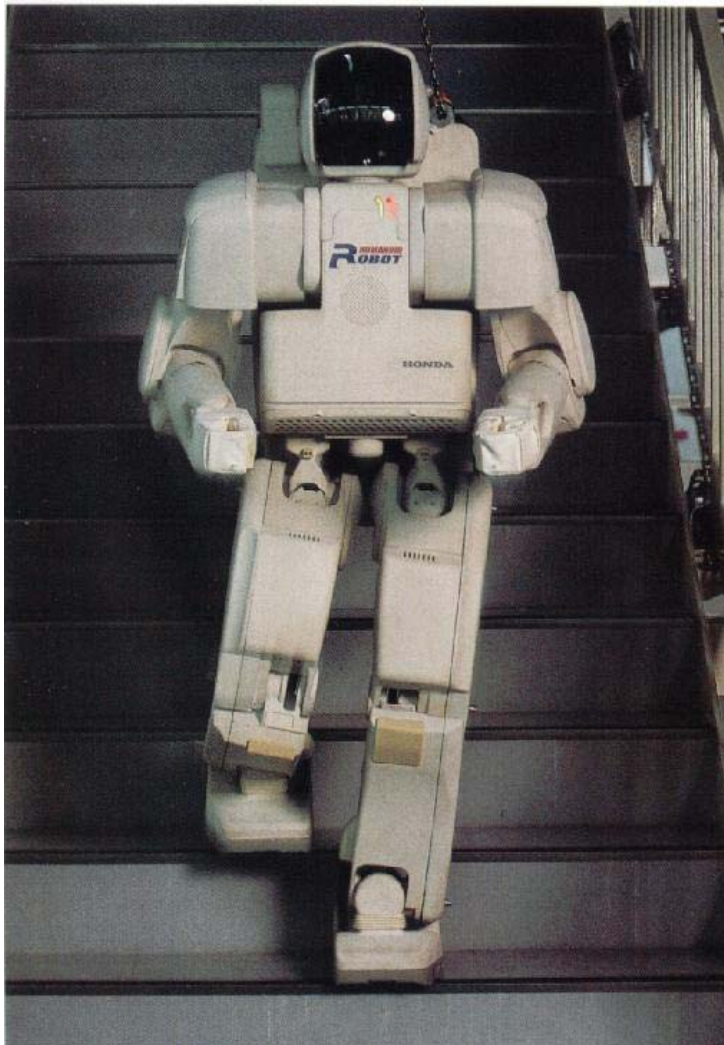
1970-Shakey the robot reasons about its blocks

Built at Stanford Research Institute, Shakey was remote controlled by a large computer. It hosted a clever reasoning program fed very selective spatial data, derived from weak edge-based processing of camera and laser range measurements. On a very good day it could formulate and execute, over a period of hours, plans involving moving from place to place and pushing blocks to achieve a goal.

Courtesy of SRI International

The Honda Humanoid Robot





Honda

Brooks' alternative

Brooks' alternative is in terms of many individual and largely separate **behaviours** – where any one behaviour is generated by a pathway in the 'brain' or control system all the way from Sensors to Motors.

No Central Model, or Central Planning system.

Sojourner Rover on Mars

NASA and JPL

July 1997

Based heavily on Brook's ideas

"Fast cheap and out of control!"

Sojourner

Sojourner

Semi-autonomous

Signals from Earth took around 30 minutes to reach Mars

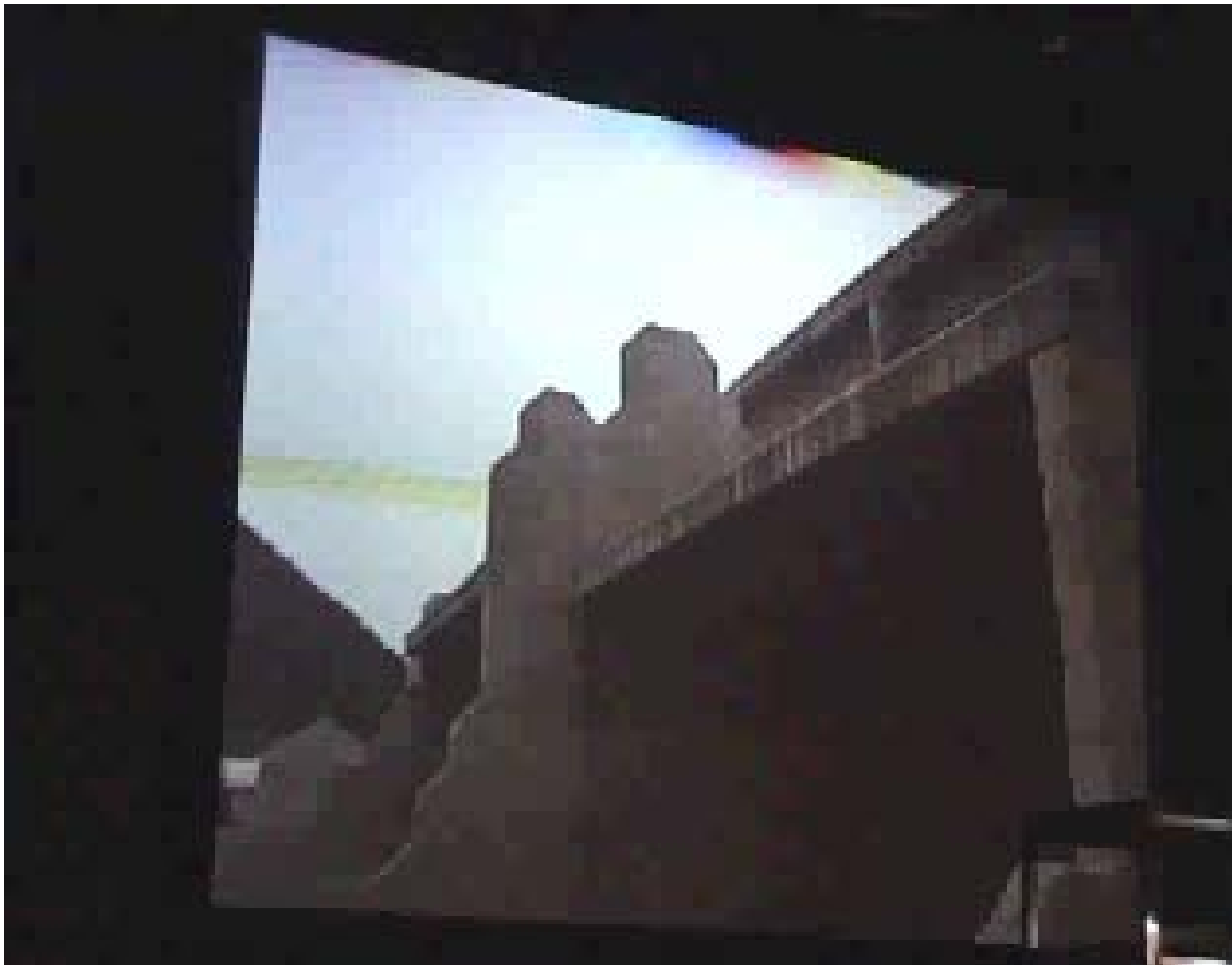
Subsumption summary

- New philosophy of hand design of robot control systems
- Incremental engineering – debug simpler versions first
- Robots must work in **real time** in the **real world**
- Spaghetti-like systems unclear for analysis
- Not clear if behaviours can be re-used
- Scaling – can it go more than 12 behaviours?

Over rough terrain

For military or rescue

For military or rescue



Underwater

THE END !