University of BRISTOL

More Advanced GA Concepts

Seth Bullock

Contents

This lecture covers a number of Evolutionary Computation concepts:

- Premature Convergence
- Neutrality
- Quasi-Species
- Epistasis
- Problem Modularity
- No Free Lunch
- Evolvability

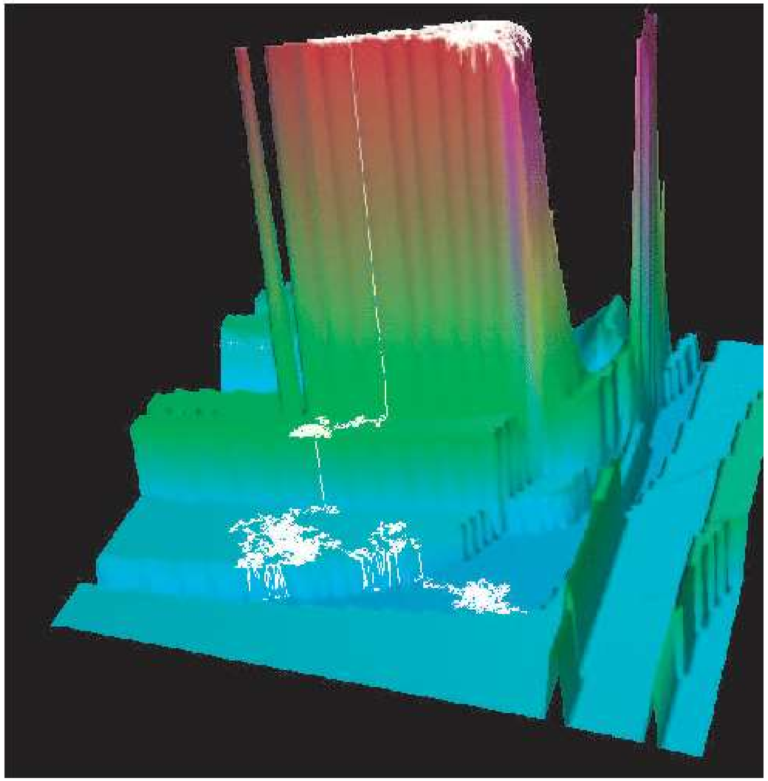
The Problem of Local Optima

- Hill-climbers get stuck on local optima; Exhaustive search does not
- GAs are less likely to get stuck on local optima than hill-climbers:
  - …because they maintain and evolve a *population of solutions*
  - …because genetic operators may generate a *wider range of neighbours*
  - These reasons exploit and rely on *diversity* in the evolving population
- Local optima are still a problem as *pop diversity is not guaranteed*
- Should a population of solutions end up *converged* within one basin of attraction they can find it difficult to escape
  - Getting stuck like this is called: *premature convergence*

University of BRISTOL

Neutrality

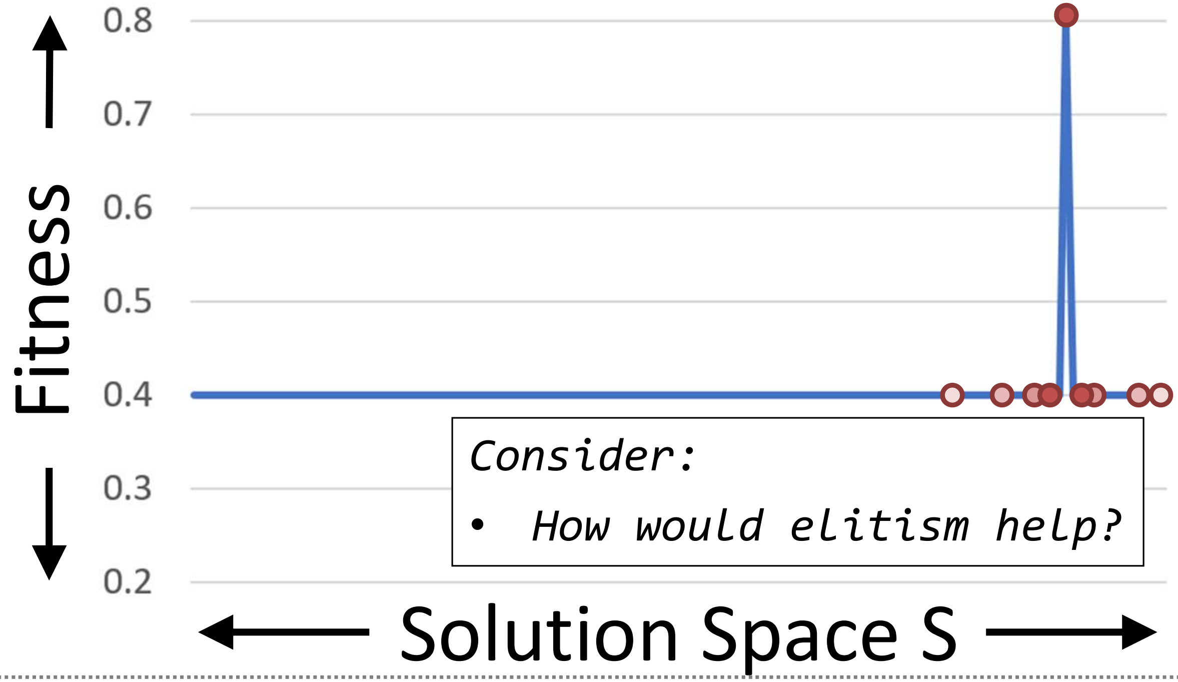
- Neighbouring genotypes with equal fitness are ‘selectively neutral’
  - …even if their phenotypes are different.
  - Evolution cannot choose between them…
  - …which results in ‘evolutionary drift’
- Neutrality is often easy to overlook and hard to visualize, but it can be key…
  - E.g., ‘neutral networks’ that percolate the search space may allow converged populations to escape “local optima”



Barnett (2002).
Explorations in Evolutionary Visualisation

The Invisible Needle

- Recall the "needle" fitness landscape we saw in a previous lecture
- The needle is hard to find because there's no gradient to climb
- But even if we start the pop *on the needle*, we still might not find it…
  - Notice: it's very rare for an offspring to be an exact copy of their parent
  - So a perfect solution will tend to have unfit offspring…



The Quasi-Species Concept

- Manfred Eigen & Peter Schuster developed the *quasi-species* concept to explain what's happening here:
  - High GA mutation rates (much higher than for DNA) mean that the evolving population is a *cloud* of points on the landscape: a *quasi-species*
  - If the selection pressure on the population to reward good solutions…
  - …is overcome by the mutation pressure that constantly corrupts them…
  - …then the population effectively cannot even "see" the needle.
- Conversely: if an evolving population *is* able to find a good solution, *and stay there*, we can infer that the solution must be surrounded by other pretty good solutions – it is *robust*

Epistasis

- If we're very lucky, fitness is a *linear function* of the gene alleles:
  - The contribution of each gene to fitness is *independent* of other genes
  - `1-max` is a linear fitness function: fitness = the number of 1s in a bitstring
  - Trivial to optimize: optimize each gene independently
- But almost always, fitness is a *non-linear function* of gene alleles:
  - The contribution of genes to fitness is *interdependent* to some extent
  - The best allele choice at one gene *depends* on the alleles at other genes
  - E.g., baby skull size ↔ size of birth canal; weapon range ↔ visual range
    - This is *epistasis*: more epistasis makes a search problem harder

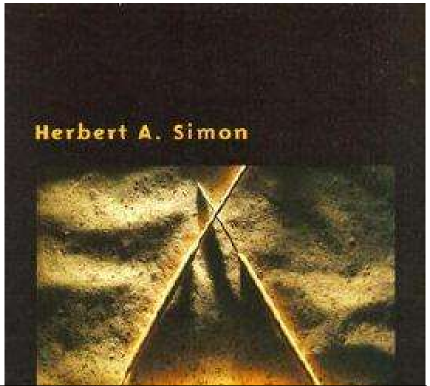
University of BRISTOL

Epistasis

- The presence of epistasis ⇔ the existence of local optima.
  - Epistasis / local optima are different ways of describing the same thing:
  - Non-linearities in the mapping of fitness onto the search space
- Epistasis means we can't optimize each gene *independently*
  - Instead our algorithm must *co-optimize* the alleles of multiple genes
- How big is the set of interacting genes? (the 'order' of interaction)
- How many sets of interacting genes? How do they overlap?
- More generally, what is the *structural modularity* of the problem?

Modular Decomposability

- Herb Simon introduces a nice way of thinking about problem modularity in his book *The Sciences of the Artificial* (1969): trying to crack a safe
  - A bank safe has *N* dials, each with 100 different settings
  - Only *one* setting is correct on each dial
  - *All* dials need to be correct to open the safe
  - (Recall that Dawkins uses the same example in his *Horizon* episode…)
      - How hard is it to open the safe?



Modular Decomposability

The answer depends on the *modularity* of the safe

- For a safe with perfectly silent dials: we must search all 100^N possibilities
  - We can expect to have to check ½ of them => $100^N/2$ to crack the safe
  - No modularity; no useful problem structure; solution has strong epistasis
- But if each dial gives a little *click* only when it is at the correct setting:
  - We can expect to have to check ½ of the 100 possibilities *for each dial*
  - => $100N/2$ tries to crack the whole safe (much less time)
  - Full modularity; useful problem structure; solution has zero epistasis
    - The same as the "Methinks it is like a weasel" & 1-max problems

University of BRISTOL

Partial Decomposability

- But real problems tend to lie somewhere in between:
  - Partial modularity; useful problem structure; solution has some epistasis
  - Compare: "Methinks…" vs. evolving any correct English sentence
  - English sentences have modules (words) that depend on each other
- Watson (2006): consider a safe with dials that are sensitive to each other. For each dial, n settings (including the correct one) give a little click:
  - n=1 (full modularity); n=100 (no modularity); n=20 (some modularity)
  - We must try $n^N/2$ combinations (much less than $100^N/2$) to crack the safe
  - (After we find which are the n clicky settings on each dial: 100N tries)
    - What if n is halved for every dial that is in the correct setting?

No Free Lunches

- Wolpert and Macready (1997): No Free Lunch for Optimization:
  - When algorithm performance is averaged across *all possible problems*:
  - *Any two optimization algorithms will exhibit equivalent performance*
  - *i.e., no optimization algorithm can do better than blind random search*
- The insight here is that every search algorithm other than random search has a *search bias*: it works on a hunch, gamble, or heuristic
- For every time the hunch pays off, there's a time where it doesn't
- Perhaps need to shift focus to *satisficing* rather than optimising…
- Algorithm quality depends on the *problem structure* that it faces

Learning How to Search

- Recall that the structure of *S* (including it's modularity) is influenced by our choice of *representation* and *genetic operators*
- So, we can restructure the space by making different choices
- Good choices could be the difference between success and failure
  - Might it be possible to learn which operators to use when?
  - Or to learn a good genetic representation?
  - ..either for a whole class of different, related problems (e.g., timetables)?
  - ..or *online* for one problem *as we are trying to solve it*
    - Could a GA implement evolution that gets more + more powerful?

University of BRISTOL

The Evolution of Evolvability

- Natural evolution has got more powerful over time:
  - cf. the ‘major transitions’ in evolution (Maynard Smith and Szathmáry)
  - E.g., Single=>Multi-cellularity; A-sex=>Sex; Non-Social=>Social/Cultural
- In fact, Dawkins’ paper at the first Artificial Life conference is on what his *Biomorphs* tell us about “the Evolution of Evolvability”.
  - ‘The ability of a population to generate *adaptive* genetic diversity’
  - (Here *adaptive* means ‘well adapted’ to the problem at hand)
  - i.e., evolvability is: how good is a population at evolving
    - A poor GA has low evolvability; Nature exhibits high evolvability

University of BRISTOL

The Evolution of Evolvability

- More subtle mechanisms in nature:
  - DNA repair – maintains some parts of the genome more carefully
  - Chromosome Organisation – nearby genes less likely to be disrupted
- Genetic Algorithms can look to exploit similar tricks:
  - Analyse how fitness varies in the current population – build a model
  - Use this model to predict what type of genotypic variation will be best
  - Effectively learning the structure of the problem during evolution
- E.g., “How can evolution learn?” (Watson & Szathmáry, 2015)
  - http://www.bbc.com/earth/story/20170301-life-may-actually-be-getting-better-at-evolving

Example Questions

- Complete the missing fitness entries in the two tables of genotypes and their associated fitness values, below, such that there is (i) no epistasis, (ii) some epistasis. *[2 marks]*
- Give an example of two aspects of a real-world problem that are linked epistatically. Explain your answer. *[4 marks]*
- A problem comprises two complex but almost independent sub-problems. How should a GA's genotype be structured, and what type of crossover should be used?*[6 marks]*

University of BRISTOL

Thank you!