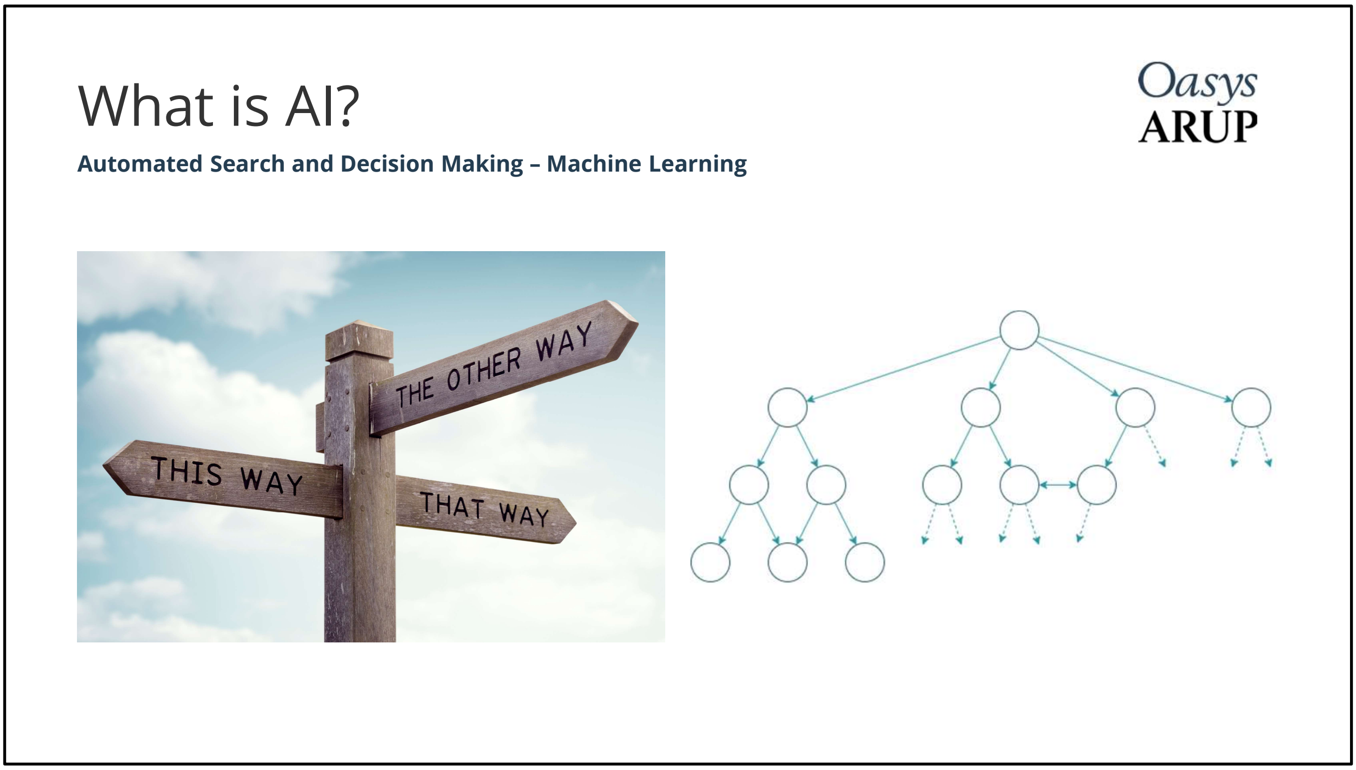
Oasys
ARUP

Optimisation == Machine Learning

IABSE Future of Design

Peter Debney BEng(hons) CEng MBCS FIStructE
21 09 2023



What is AI?

Automated Search and Decision Making – Machine Learning

Oasys
ARUP

While AI was first intended to replicate human thinking, it became a series of methods to create intelligent behaviour – making a computer find answers that cannot be directly calculated.

It is not possible to directly solve the question: “what is the shortest route between two points”. Instead, AI research produced algorithms such as A* to intelligently search for the shortest route with a mixture of estimates, calculation, and data storage.

Searching through unsorted databases is also an AI application, the most famous of which has become a verb: To Google. Google’s search rankings were initially based on keywords and webpage connectedness to predict what the best order was to present the results but has subsequently added machine learning to record and use what links people actually click on.

It is often forgotten that optimisation is a form of artificial intelligence – machine learning, search for and choosing the best answers from a myriad of options. Optimisation uses a large number of methods (see my Structural Engineer paper from a few years ago on the subject), many of which are inspired by nature: the evolution of life, how that life works together to find food, how bones react when they are exercised, or what happens when metal cools.

Why Optimise?

Oasys
ARUP

Construction has a major impact on the environment.
As an industry we are responsible for about 8% of the world’s greenhouse gasses.
Sand and gravel is in such demand for concrete that organised crime is now a major supplier of raw materials – entire beaches and islands have been stolen and illegal dredging leads to flooding and habitat loss.
Water is in even more demand, triggering about 200 violent conflicts globally in the last decade.

If we are to build then it must be using less materials, which must have a lower environmental impact.

Clients want their designs to be as cheap as possible, while still maintaining standards and safety.

Plus, we are engineers: we like the technical challenge of achieving the goal with the least structure.

We must do more with less.

Structural Optimisation

Oasys
ARUP

Minimise one or more property while satisfying several constraints:

Find the best solution among the set of feasible solutions

Feasible solutions = Design Space

Topology Optimisation
Shape Optimisation
Section or Size Optimisation

This means that you need:

- An objective – a quantitative measure of the performance of your system or model
  - Weight / Cost / Stiffness / Environmental Impact
- Variables – the values that change to achieve the objective
  - Section size / nodal locations / number of beams / …
- Constraints – when the variables are acceptable
  - Allowable stresses / section strength / deflection limits / …

Section or **Size Optimisation**

e.g. Choose the best section sizes to achieve the objective

Shape Optimisation

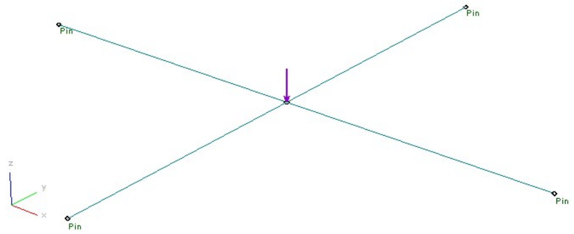
e.g. Modify the joint or connection locations of the structure, or the shape of the members

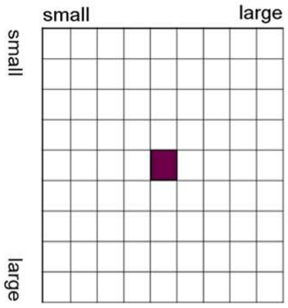
Topology Optimisation

e.g. choose how many elements there are in a truss, how many columns, etc.

NB the boundaries between these types of optimisations can be fuzzy

Searching the Design Space





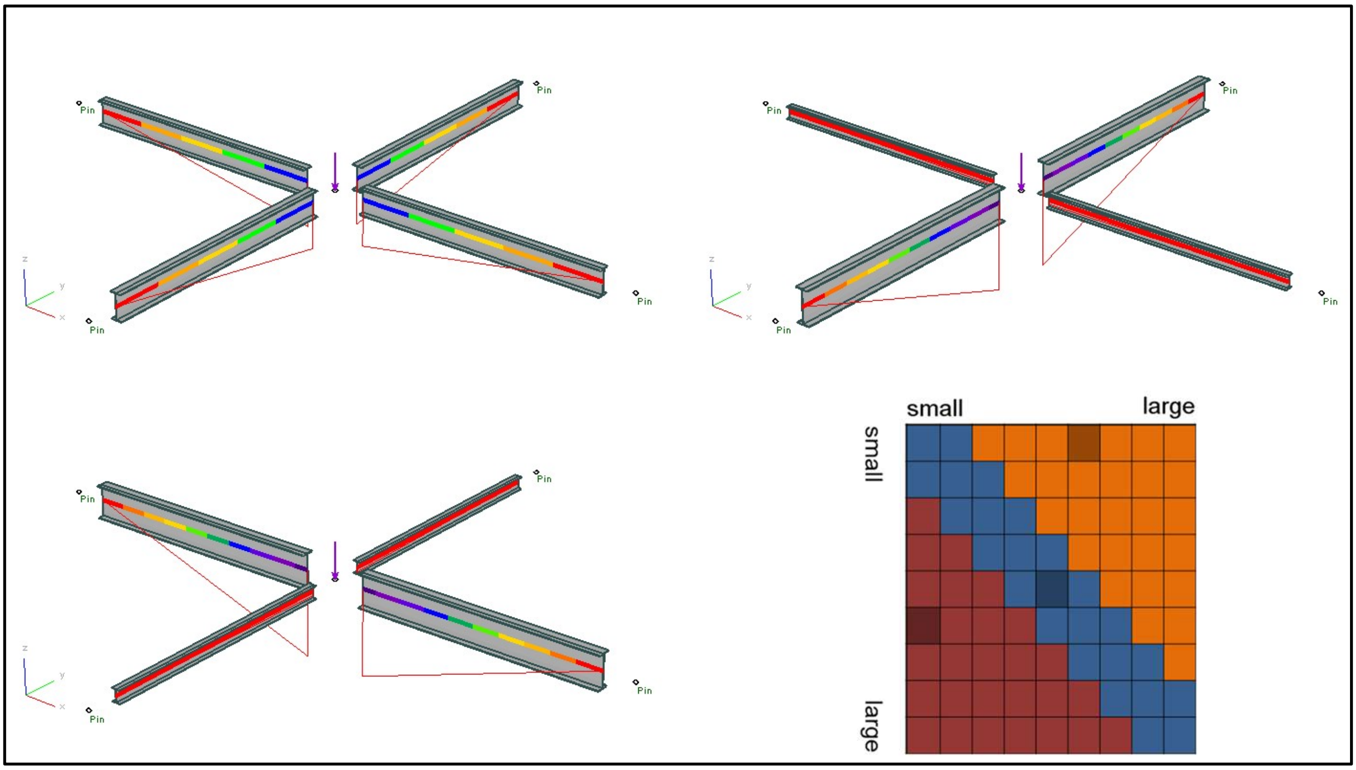
How does optimisation relate to machine learning? We cannot (normally) directly determine the best designs, so instead we have to search for them, learning what works as we go along. Unlike the machine learning as described in the press, which usually uses deep neural networks to learn from huge databases, optimisation learns from the analysis model itself, changing it in the process.

To illustrate the concept, let's look at a simple example: a crucifix of steel beams, all the same length and all fully moment connected, with a central point load

If we consider all the possible range of beam sizes for the two directions, we can plot them as a grid.

This is known as the design space: the range of possible design alternatives.

If we decide that we are going to build this structure with two particular member sizes, then they can be plotted as a particular location within this design space



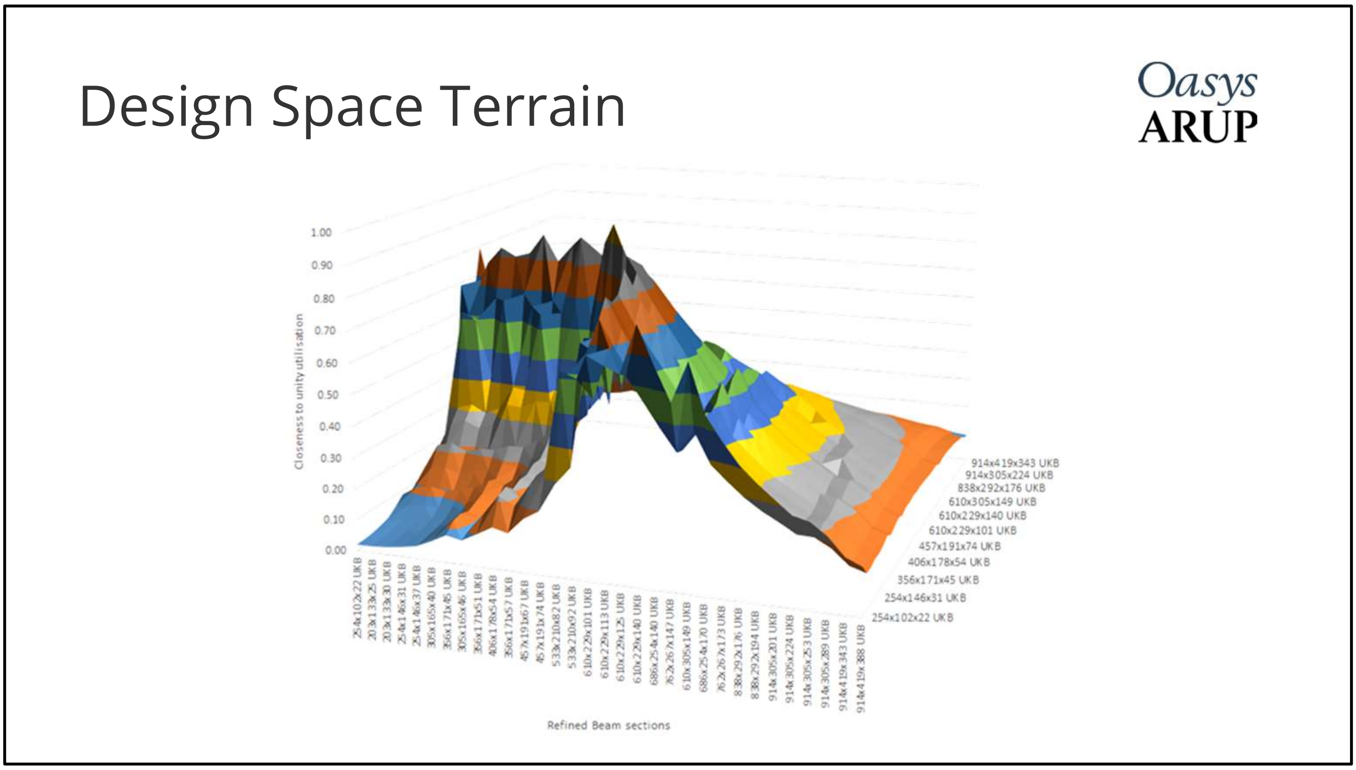
If the if beams are the same size, then they will attract the same loads and hence both beams will work equally hard.

But the astute will have noticed that this is only one possible solution to this problem:
If one beam is particularly larger than the other then it will attract all the load, leaving the other to simply be a mid-span restraint.

Typically, we might solve this problem by picking two section sizes, analysing the system then designing the beams (choosing the section size that carries the load). We then need to repeat as the changes to the sections will change the distribution of the moments and forces, which might need another change to the sections.

Going back to the design space, we note that the solution found depends on where we start in the design space. This gives each solution a tributary area

With this simple example the design space is also relatively simple but note that there are about 100 universal beam sizes available in the UK market, giving a design space of approximately 10,000 options.

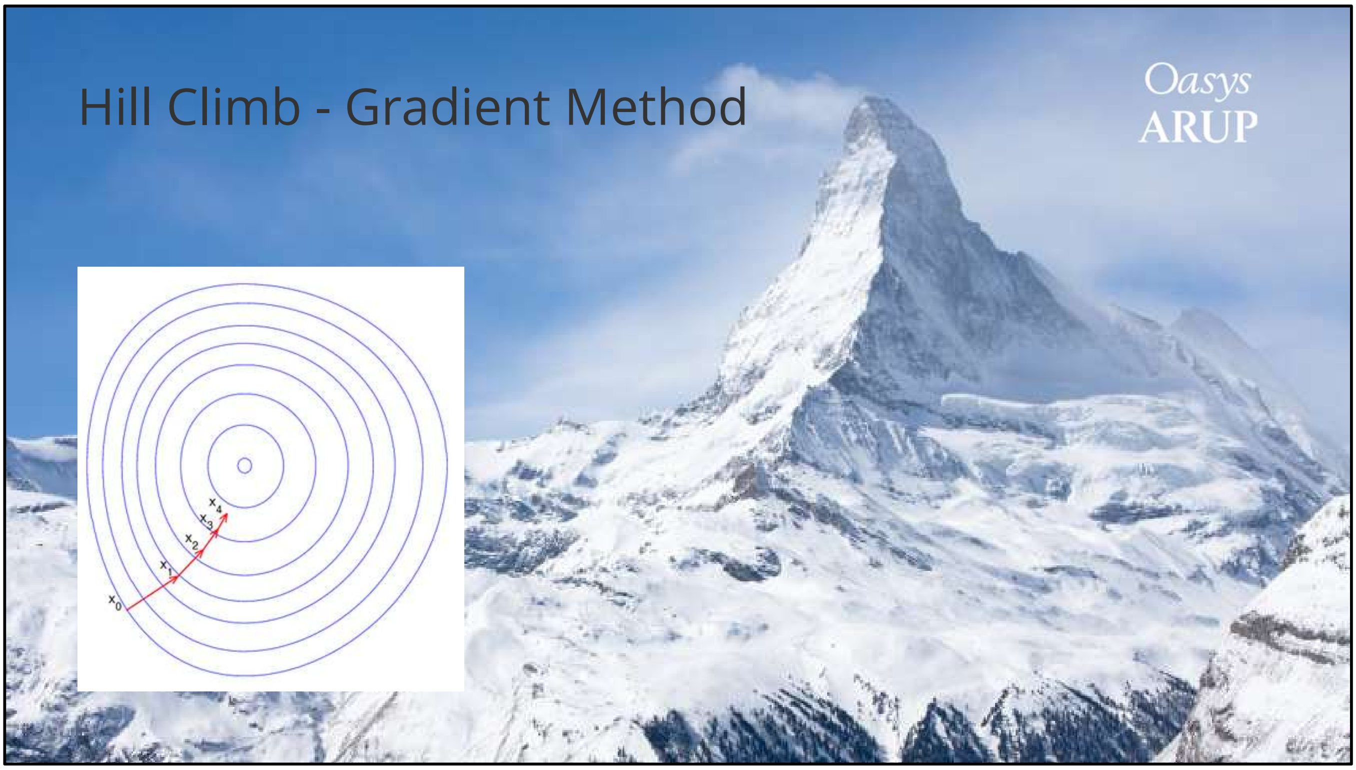


Design Space Terrain

This is a view of the actual design space with the elevation set as the solutions' fitness

As you can see there is a definite best point, but there are also a range of good solutions as well (which are asymmetrical)

How might we search this space for the optimum answer?



Hill Climb - Gradient Method

Oasys
ARUP

Gradient methods are the simplest way to search the design space for the optimum solution, and they tend to be the fastest, and the most brittle. They either work well or not at all.

Essentially, they work either by determining the local gradient and then heading up (or down as appropriate) or just test the adjacent locations to see which is the best, then carry on from there. They stop when they cannot find a better point, but this is not guaranteed to be the peak, but could just be a local blip – a foothill.

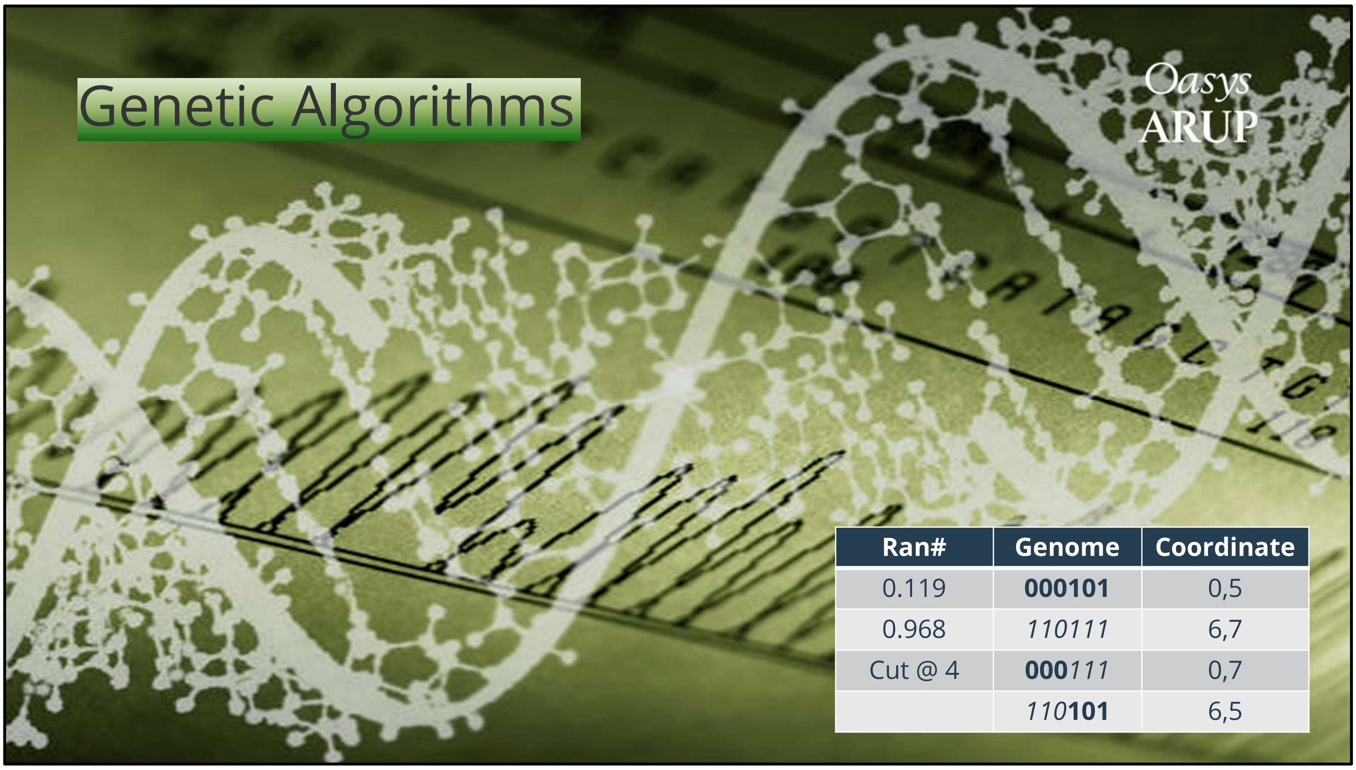
Simulated Annealing

Oasys
ARUP

Simulated annealing looks to get around gradient methods tendency to find sub-optimal answers by adding in a heat variable. Like annealing of metals, the behaviour changes as things cool.

In this instance the heat variable sets the distance the point of interest jumps through the design space. If the new point is better, it moves to there and increases the heat. If it is worse, then it turns the heat down and tests another closer random point.

The idea is that the jumps are large at the start and gradually get smaller and smaller as it converges on the solution.



Genetic Algorithms

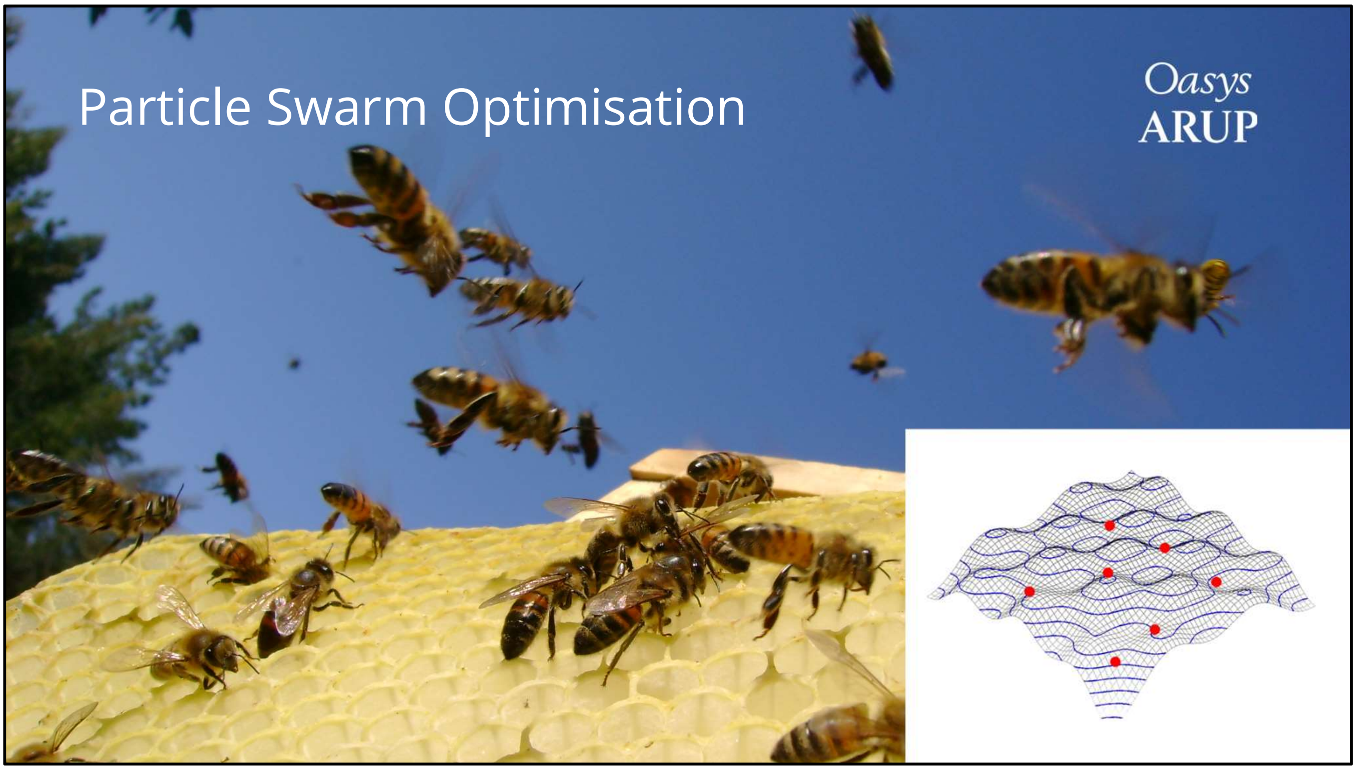
Oasys
ARUP

Ran#	Genome	Coordinate
0.119	**000101**	0,5
0.968	*110111*	6,7
Cut @ 4	**000***111*	0,7
	*110***101**	6,5

Unlike Hill Climb and Simulated Annealing, Genetic Algorithms work with multiple points of interest in the design space, tests each one, then cross-breeds the best to give new locations in the design space to test.

It does this by treating the coordinate of each point in the design space as a DNA string, which can then be split and joined with another.

GAs can be very effective in difficult design spaces but are notoriously slow



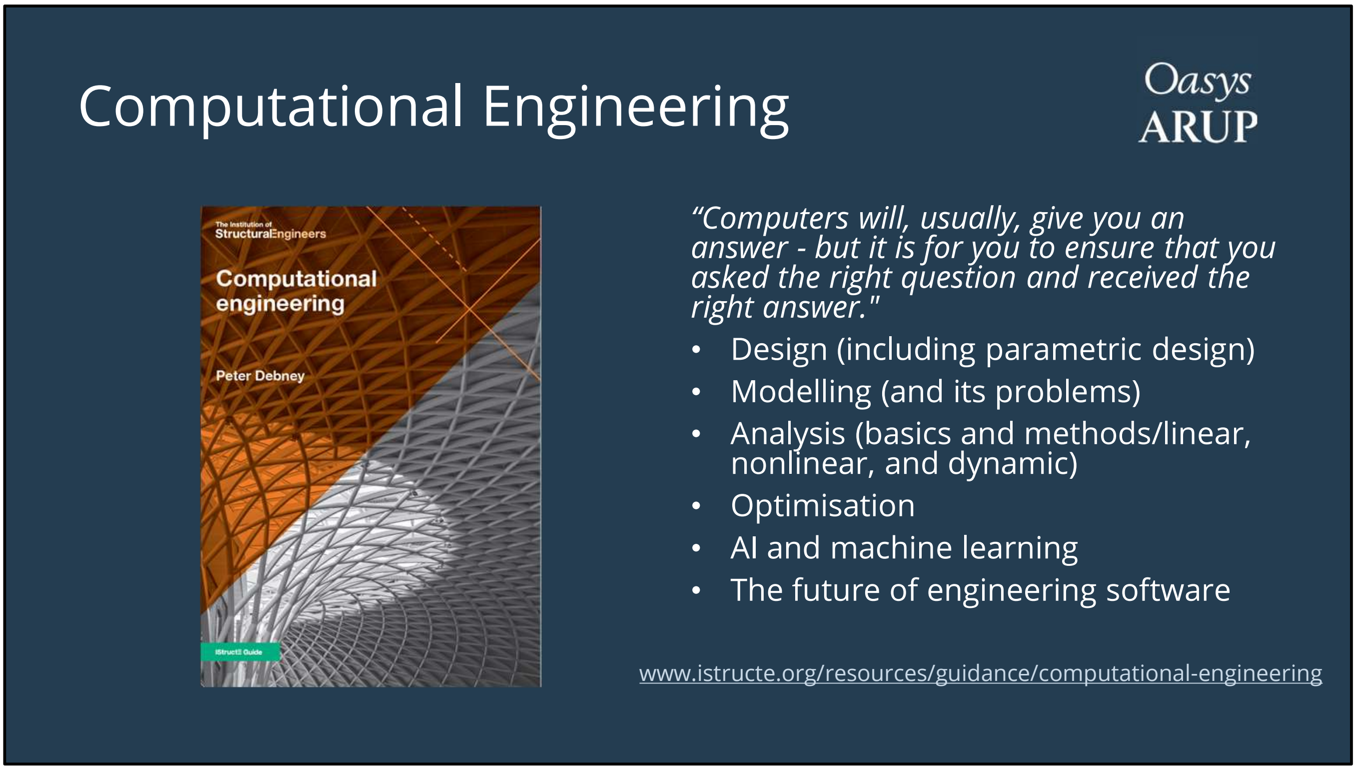
Particle Swarm Optimisation

Oasys
ARUP

One of my favourite methods is the particle swarm, which is inspired by the swarming of bees or flocking of birds.

The design space is explored by a swarm of “particles” and the design space is tested at their locations. The particles have a velocity, which is adjusted based on the direction of the location of the best design found by that particle and that of the best design found overall.

This means that the particles learn from each other and one or two focus in on the optimum while others explore the immediate area.



Computational Engineering

“Computers will, usually, give you an answer - but it is for you to ensure that you asked the right question and received the right answer."

- Design (including parametric design)
- Modelling (and its problems)
- Analysis (basics and methods/linear, nonlinear, and dynamic)
- Optimisation
- AI and machine learning
- The future of engineering software

www.istructe.org/resources/guidance/computational-engineering

This is a very brief exploration on how we can use machine learning for optimisation. I go into more detail in my book “Computational Engineering” – please excuse the commercial plug – which also explores other methods and useful aspects of using computers for structural engineering.

You can get it from the IStructE website.