Evolutionary Computation

COMP2002

Lauren Ansell

Introduction

Today's topics:

- Optimisation
- Evolutionary computation
- Genetic algorithms

Session learning outcomes - by the end of today's lecture you will be able to:

- Recognise optimisation problems and describe their simple characteristics
- Interpret the steps of the genetic algorithms

Optimisation

Optimisation is a mathematical method to select the best element from a group of available alternatives.

In the selection process there may be some rules (criterion) which need to be followed.

There are two subfields in optimisation:

- discrete optimisation
- continuous optimisation

An optimization problem consists of maximizing or minimizing a real function by systematically choosing input values from within an allowed set and computing the value of the function.

Travelling Salesperson Problem

The task: a salesperson must visit cities A, B, C and D in one go – what is the most efficient route they should take?

The problem is fairly simple to solve for four cities – just enumerate each possible route and calculate its efficiency

What if there are 20 cities?

There are a total of $N!$ possible routes

That is $20 \times 19 \times 18 \times \ldots \times 3 \times 2 \times 1 = 2,432,902,008,176,640,000$ routes

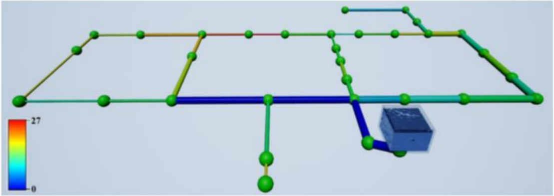
Container Stowage Problem

A modern large container ship can carry over 20,000 containers

How should the containers be loaded?

What problem constraints might exist?

Water Distribution Network Design



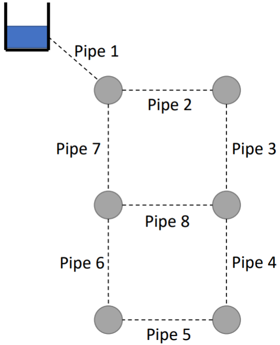
M. B. Johns, D. J. Walker, E. Keedwell and D. Savić. "Interactive Visualisation of Water Distribution Network Optimisation". EPiC Series in Engineering (3):995-1003, 2018.

Design new WDNs or upgrade existing networks.

Optimisation of pressure, water quality, adherence to pressure constraints, robustness of the network. . .

TwoLoop problem

Identify the optimal size for
eight pipes (all pipes have equal length).
14 possible
pipe diameters (1-24 inches) each with
a known unit cost ($/m) – lookup table



Solution

Pipe #	1	2	3	4	5	6	7	8
$x =$	3	8	10	10	22	6	1	24

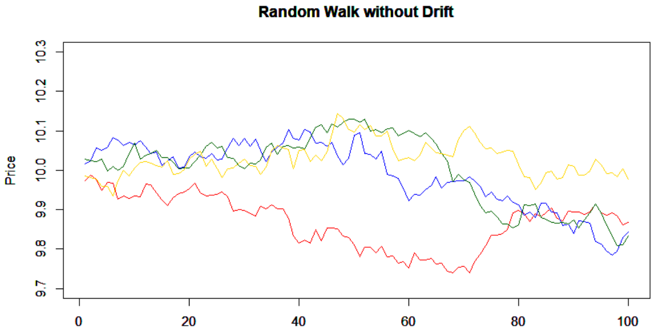
Fitness function is sum of pipe costs

$$y = f(x) = 8 + 23 + 32 + 32 + 300 + 16 + 2 + 550 = \$963/\text{m}$$

Random Walk

Simplest optimisation algorithm – not intelligent

Generate a solution at random.
Then for a number of iterations, generate further random solutions.
Across the iterations keep track of the best solution found
Eventually the algorithm converges – but extremely slowly. . .



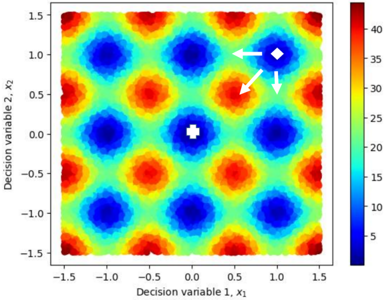
Hill Climber

Generate a random solution x.

Perturb the solution – move it to a different part of the search space to produce a new solution.

If the new solution is better keep it – otherwise move back to the original solution.

The white diamond is in a local optima – reasonable quality but not the best and must get worse to get better by reaching the global optima

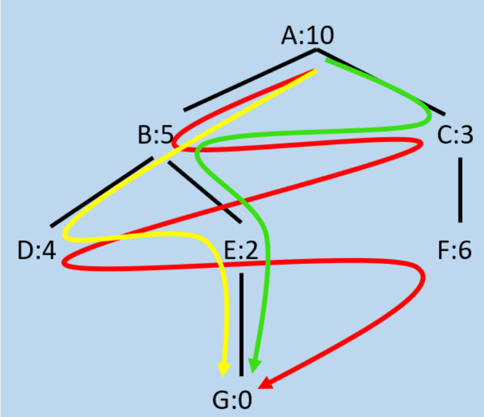


Greedy Search

Expands the node with the best score.

The best node is removed from the list.

Method is dependant on the heuristic function.



Pros And Cons Of Hill Climbers

- Hill Climbing is a simple and intuitive algorithm that is easy to understand and implement.
- It can be used in a wide variety of optimization problems.
- Hill Climbing is often very efficient in finding local optima.
- The algorithm can be easily modified and extended to include additional heuristics or constraints.

- Hill Climbing can get stuck in local optima, meaning that it may not find the global optimum of the problem.
- The algorithm is sensitive to the choice of initial solution, and a poor initial solution may result in a poor final solution.
- Hill Climbing does not explore the search space very thoroughly, which can limit its ability to find better solutions.
- It may be less effective than other optimization algorithms for certain types of problems.

Offspring generation

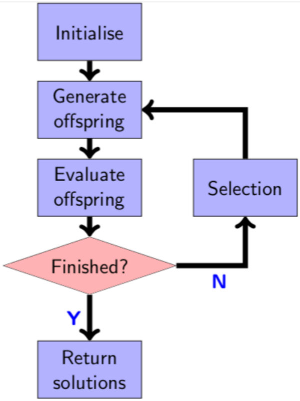
Crossover – combine
genes from two or more parents
Mutation – randomly
modify genes on a solution

Finished?

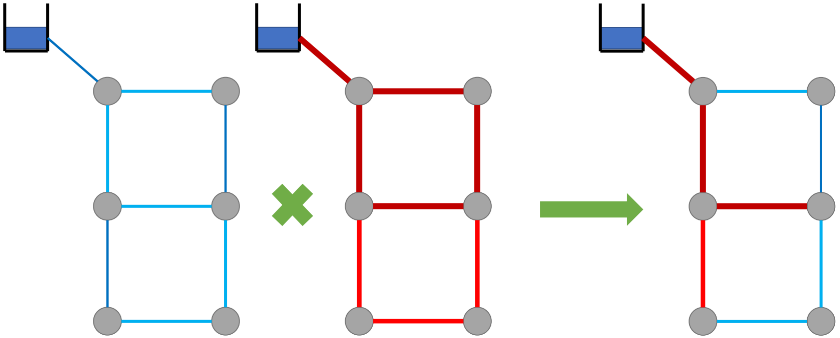
Fixed budget of function evaluations
Online convergence detection

Selection

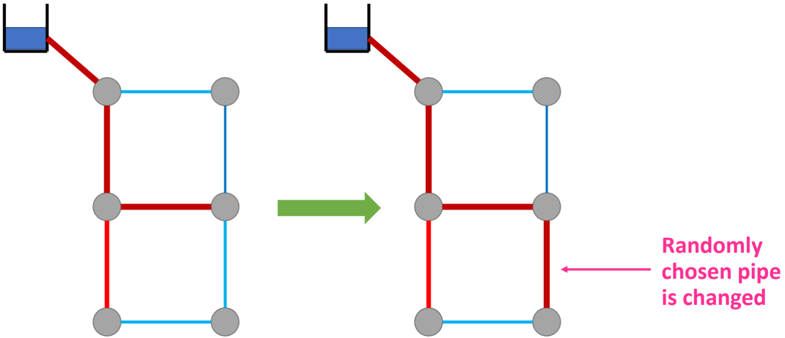
Elitist
Rank, tournament selection. . .



Crossover



Mutation

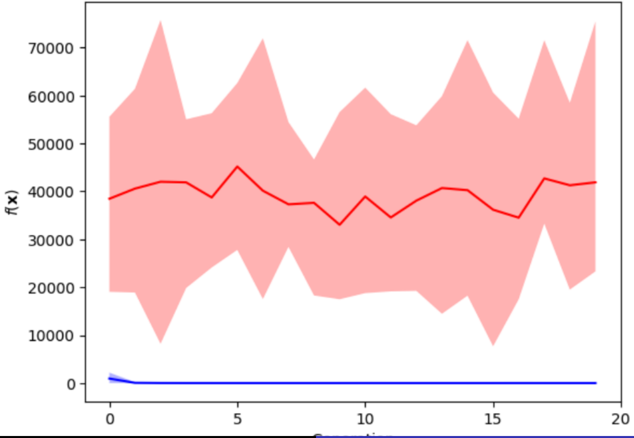


TwoLoop With A GA

Optimise TwoLoop network

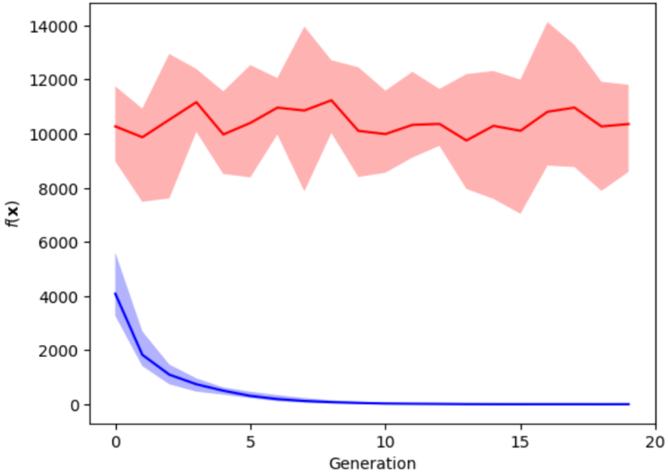
Use EpaNET to model hydraulics

Minimise cost + constraint violations



Hanoi With A GA

Same problem – different network

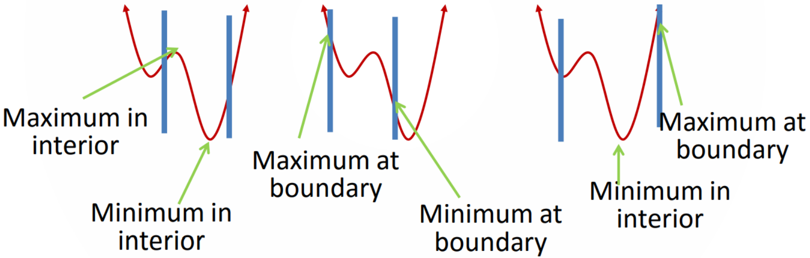


Optimising With Constraints

So far we have not discussed the limits or boundaries for the functions being fed into the algorithms.

A constraint is a hard limit placed on the value of a variable, which prevents us from going forever in certain directions.

With nonlinear functions, the optimum values can either occur at the boundaries or between them.



Examples Of Constraints

If you are attempting to maximize the objective function, typical constraints might involve time, money, and resources. The amounts of these things are limited, and these limits also place limits on the best possible value of the objective function.

If you are attempting to minimize, the constraints are more particular to the situation.

Writing Constraints

This is going to involve algebra.

Example

If a single apple costs 45p, then it follows that two cost 90, five apples cost £2.25.

In general, *a* apples cost 45a or 0.45a.

If you buy *a* apples at 45p and *b* blackberries at £2, then your total cost will be $0.45a + 2b$.

If you only have £10 to spend at the supermarket, then your total cost must be

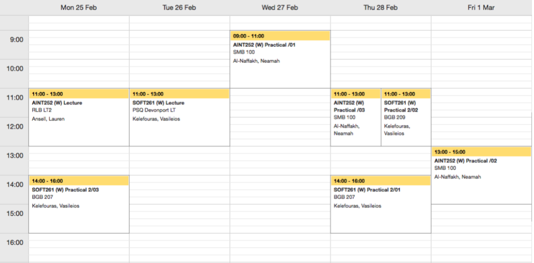
$$0.45a + 2b \leq 10.$$

Task - Writing Constraints

1. A batch of cookies requires 150g of flour, and a cake requires 200g. Write a constraint limiting the amount of cookies and cakes that can be made with 3kg of flour.
2. Box type 1 can hold 20 books and box type 2 can hold 12. Write a constraint for the number of boxes needed in order to box up 100 books.
3. If it takes you 4 minutes to bike a mile, 9 minutes to run a mile and 14 minutes to walk a mile, write a constraint that limits how many miles of each type of exercise you can get in a 45-minute lunch break.

Timetabling – Optimising With Constraints

Constraints are rules within the problem formulation that cannot be broken.



Example timetabling constraints:

- A student cannot be timetabled for two sessions at the same time.
- Two sessions cannot be placed in the same room at once.
- A session's allocated room must have capacity for the total number of students.

Evolutionary Machine Learning

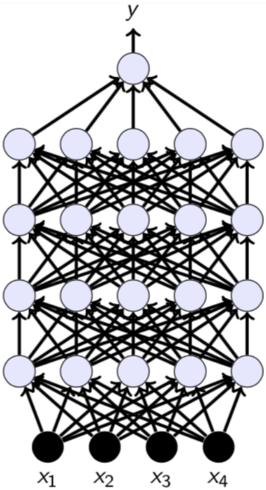
**Optimise the
weights and structure of an ANN**

What are the edge weights?

Which
nodes and edges are needed?

What
should their activation function be?

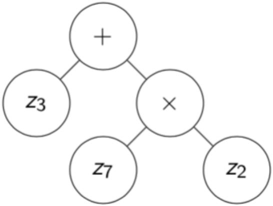
Fitness function is model accuracy



Genetic Programming

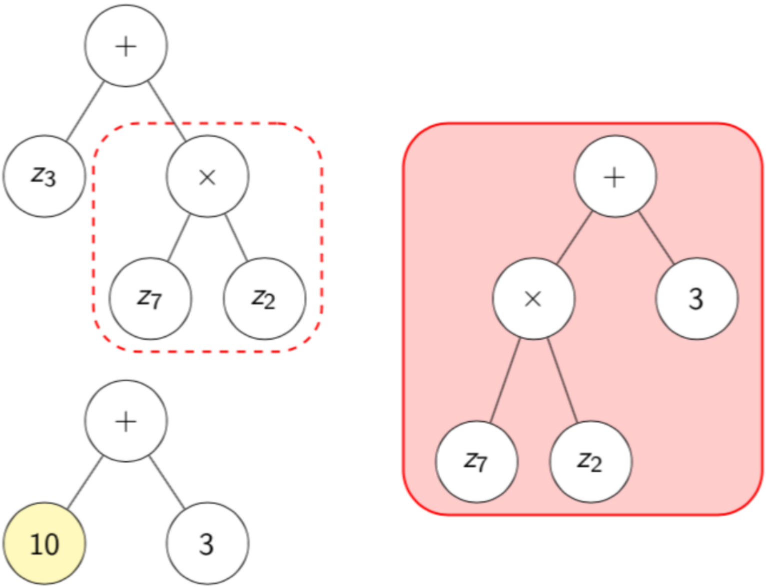
Evolve symbolic expressions or executable code with desirable properties.

Wide range of applications – modelling, gameplay controllers, search-based software engineering... Solution is represented with a tree



Given model input vector z, model output is $z_3 + (z_7 \times z_2)$

Crossover In GP



Expensive Optimisation

What if a function evaluation takes minutes, hours or even days?

Parallel evolution

Execute (e.g.) each function evaluation on a different core.
The whole EA can be run in parallel.

Surrogate modelling

Build a model that simulates the function evaluation.
The execution of the model will be much faster.
Need data on which to base the model training.
Still need to run the function evaluation at some point.

Summary

Evolutionary computation – nature-inspired optimisation

- Problem formulation
- Genetic algorithms
- Genetic programming
- Expensive optimisation