Multi-objective Optimisation

COMP2002

Lauren Ansell

Introduction

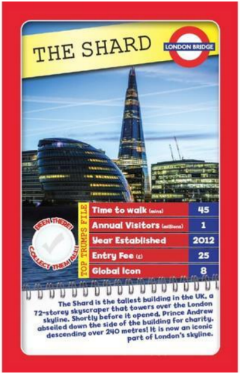
Today's topics:

- Multi-objective optimisation
- Many-objective optimisation

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

- Give examples of multi-objective optimisation problems
- Distinguish between multi-objective solutions using dominance

Top Trumps



Which criterion to choose?

What does this have to do with optimisation?

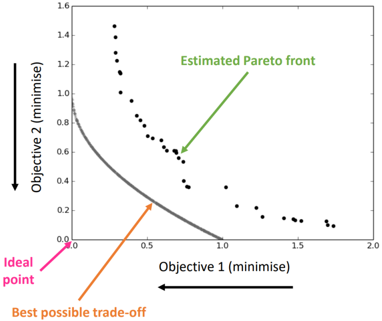
Multi-objective Optimisation

Often solution quality
is described by multiple objectives.

These solutions
are in conflict – a good solution
according to one objective is a
poor solution according to the other.

Multi-objective optimisation
– two or more objectives.

Find an estimate of the Pareto front.

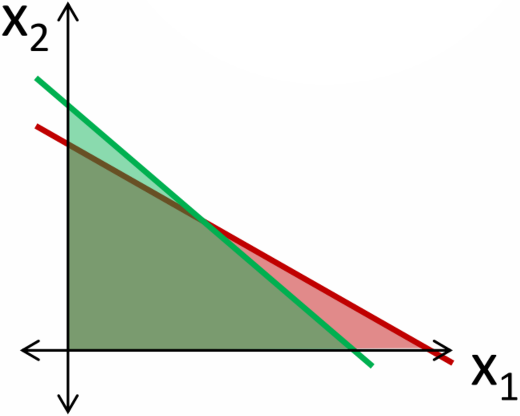


Linear Example With 2 Constraints

Suppose you have two constraints as follows:

$$2x_1 + 3x_2 \leq 34$$

$$3x_1 + 5x_2 \leq 54$$



Multi-objective Water Network Design

Minimise network cost

$$C = \sum_{i=1}^{N} c(D_i) \times L_i$$

where:

- $c(D_i)$ is the cost of the pipe of diameter D_i
- L_i is the length of the pipe

Maximise hydraulic performance

$$I_n = \frac{\sum_{j=1}^{nn} C_j Q_j (H_j - H^{req})}{\sum_{k=1}^{nn} Q_k H_k + \sum_{i=1}^{npu} \frac{P_i}{\gamma}}$$

Single-objective EAs For Multi-objective Problems

EAs rely on fitness to facilitate selection – how should we deal with multiple objectives?

Compute a single fitness value by aggregating the objectives

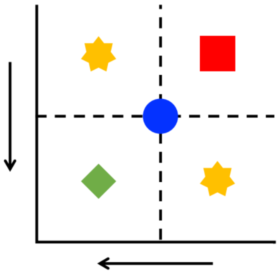
$$f(x) = \sum_{m=1}^{M} f_m(x) \cdot w_m \qquad f(x) = (f_1(x) \cdot w_1) + (f_2(x) \cdot w_2) \quad (1)$$

Issues with aggregated objectives

- How should the weights be chosen?
- Many different runs needed to approximate the trade-off between objectives
- Reduced search ability – concentrating on a specific part of the space as governed by the choice of weights

Pareto Dominance

One solution
dominates another if it is no worse
than the other on any objective
and better on at least one objective.



Assuming minimization:

- Green dominates all solutions
- Blue dominates red
- Blue and the yellows
  are mutually non-dominating
  (neither dominates the other,
  they are incomparable)

(1+1)—Evolutionary Strategy

Algorithm: (1+1)—ES

```
x := initialise()
y := (f_1(x), ..., f_M(x))
while not converged:
    x' := reproduction(x)
    y' := (f_1(x'), ..., f_M(x'))
    if not y dominates y':
        x := x'
        y := y'
```

Essentially a multi-objective hillclimber.

This algorithm ends with a single solution – what about the Pareto front approximation?

Evolutionary Strategy Example

Use black cherry tree data

estimate the coefficients for the linear model to predict the diameter of the tree from the height.

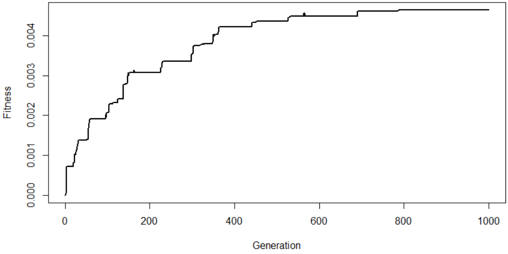
started with generating 100 potential models as the starting population.

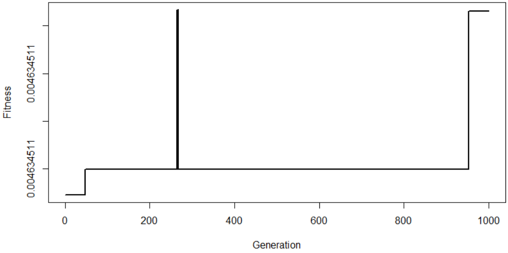
Each of the models in the starting population was evaluated for 'fitness'.

The top 10 models with the highest fitness was selected as the survivors.

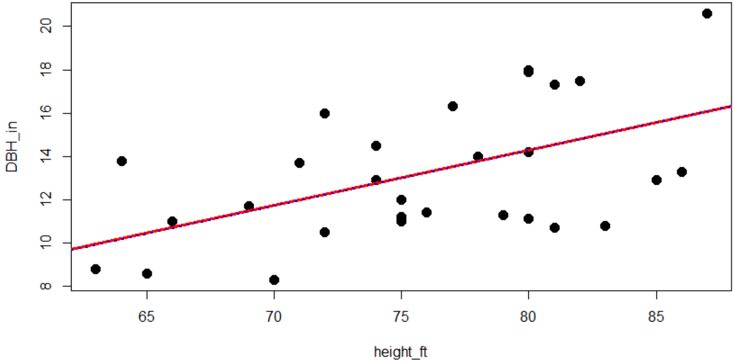
A new population of size 100 was then created from the survivors.

Results - Fitness Over Generations





Results - Final Model



Comparison of models

Evolutionary Strategy: $DBH = 0.2552542\,height - 6.151511$
Linear Model: $DBH = 0.2557\,height - 6.1884$

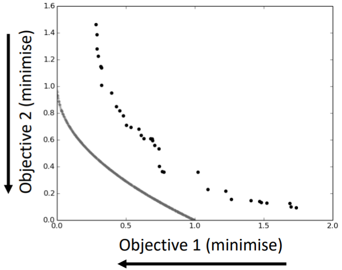
Archives

Maintain a list of all non-dominated solutions that the algorithm has found.

Each time a new solution is generated compare it to the archive

- If the new solution is dominated by the archive discard the solution
- If any member of the archive is dominated by the new solution then discard those dominated solutions

The black dots are an archive of non-dominated solutions – over time they will move closer to the black line which is the true Pareto front



NSGA-II

Non-dominated sorting Genetic Algorithm 2 – published by Deb et al. in 2002.

Widely used in a range of fields – cited 40,000+ times.

Uses a population of solutions.

Available in a large number of software packages.

Selection operator

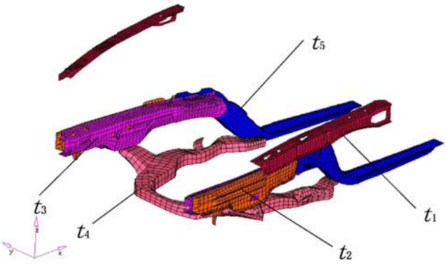
1. Rank solutions with non-dominated sorting
2. Resolve ties with crowding distance

NSGA-II Example

The vehicle crashworthiness problem is a multi-objective problem where the crash safety level of a vehicle is optimized.

A higher safety level means how well a vehicle can protect the occupants from the effects of a frontal accident.

There are five decision variables that represent the thickness of reinforced members around the car front.



Objectives And Constraints

The following is 3 objectives that need to be achieved for optimal crashworthiness:

$$\begin{aligned}\text{minimize } f_1(x) = {} & 1640.2823 + 2.3573285x_1 + 2.3220035x_2 \\ & + 4.5688768x_3 + 7.7213633x_4 + 4.4559504x_5\end{aligned}$$

$$\begin{aligned}\text{minimize } f_2(x) = {} & 6.5856 + 1.15x_1 - 1.0427x_2 + 0.9738x_3 + 0.8364x_4 \\ & - 0.3695x_1x_4 + 0.0861x_1x_5 + 0.3628x_2x_4 \\ & - 0.1106x_1^2 - 0.3437x_3^2 + 0.1764x_4^2\end{aligned}$$

$$\begin{aligned}\text{minimize } f_3(x) = {} & -0.0551 + 0.0181x_1 + 0.1024x_2 + 0.0421x_3 \\ & - 0.0073x_1x_2 + 0.024x_2x_3 - 0.0118x_2x_4 \\ & - 0.0204x_3x_4 - 0.008x_3x_5 - 0.0241x_2^2 + 0.0109x_4^2\end{aligned}$$

Constrained with:

$$1 \leq x_i \leq 3$$

Results - Table

Let's look at the optimal solution: those who belong to the first front.

f_1	f_2	f_3	x_1	x_2	x_3	x_4	x_5
1675.49	6.143	0.264	1.000	3.000	3.000	1.000	1.000
1661.71	8.305	0.071	1.000	1.000	1.000	1.000	1.000
1692.02	10.562	0.042	3.000	3.000	1.000	2.560	3.000
1692.02	10.562	0.042	3.000	3.000	1.000	2.560	3.000
1681.17	6.914	0.167	1.000	2.998	2.293	1.000	3.000
1667.38	7.006	0.111	1.000	2.995	1.216	1.006	1.002
1679.19	8.133	0.063	1.299	3.000	1.000	1.417	3.000
1684.96	9.268	0.051	2.737	3.000	1.000	1.773	2.917

We can achieve more than 1 similar objective values, indicating that all of those solutions are non-dominated to each other.

Comparison Of Solutions

Is the first solution better than the second?

The f_1 value of the first solution is higher than second solution.
The f_2 value of the first solution is lower than second solution.
The f_3 value of the first solution is higher than second solution

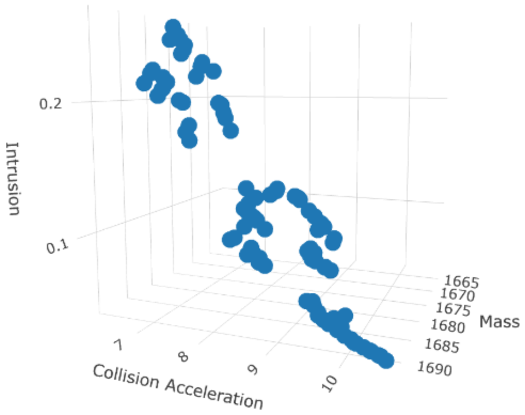
The second solution can dominate or better than the first solution if the f_2 value is lower than the first solution.

How about the first solution with the third one?

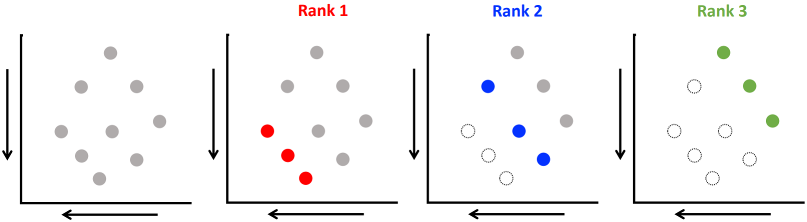
The f_1 value of the first solution is lower than third solution.
The f_2 value of the first solution is lower than third solution.
The f_3 value of the first solution is higher than third solution.

The first solution can dominate the third solution if the f_3 value is lower than the third solution.

Results - Visualize The Objective Values



Non-dominated Sorting



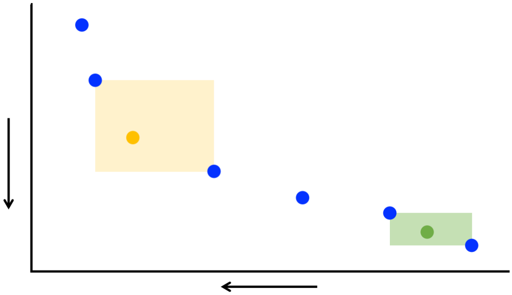
1. Identify the non-dominated solutions
2. Assign them to rank n (start with $n = 1$)
3. Temporarily discard the non-dominated solutions
4. Increment n and go to (1)

Crowding Distance – Diversity Preservation

Compute the crowding distance between neighbours in the PF approximation.

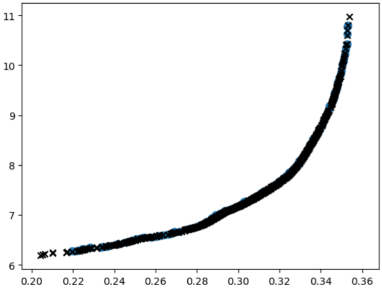
Prefer distant solutions – close solutions will be similar solutions.

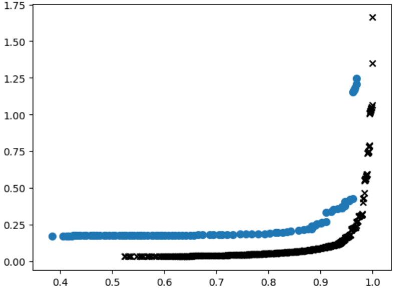
A loss of diversity will cause the search to be less effective.



The yellow solution has a greater crowding distance than the green solution so the yellow solution is preferred to the green one

Example Optimisation Runs





Black solutions are the best-known trade-off between the two objectives.

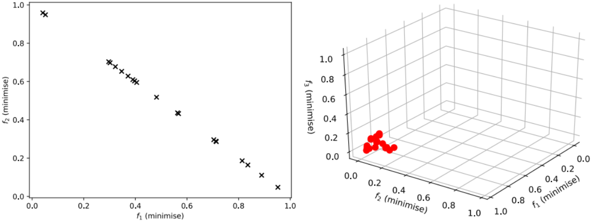
Blue solutions were found during a single run of the algorithm.

Many-objective Optimisation

Multi-objective problems are characterized by two or three objectives – real-world problems often have four or more and are called many-objective problems.

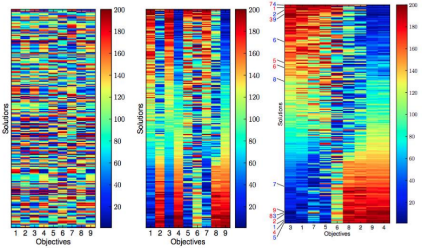
Many-objective optimisation research challenges

1. Traditional multi-objective EAs cannot search the space properly
2. Exponentially large number of solutions needed to cover the Pareto front
3. Visualising solutions for M > 3 objectives is difficult



Seriation Of Heatmaps

Enhance a heatmap visualisation – solutions are rows, objectives are columns



9-objective mutually non-dominating set of solutions to a radar waveform design problem optimised by Hughes in 2007

Compute the similarity of each solution – solve a complex optimisation problem with linear algebra that places similar solutions close together.

Works for objectives too.

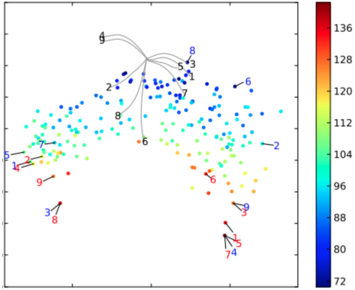
Multidimensional Scaling

Scatter plots are
useful as people understand them.

Project the M-objective solutions
into a 2D space for visualisation.

Solutions that are “close” in
Mdimensions should be close in 2D.

Incurs information loss



Summary

Multi-objective optimisation

- Many problems have two or three objectives
- Dominance for comparison
- A decision maker must select the solution to implement

Many-objective optimisation

- Optimising the problem – standard MOEAs don't work
- Visualising the Pareto front approximation