Lecture 08 – Multiobjective Optimisation

Optimisation CT5141

James McDermott

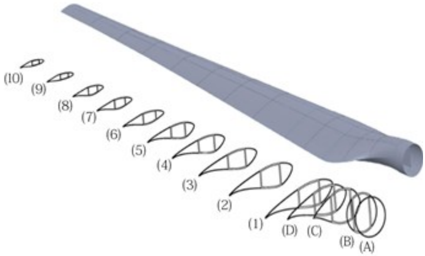
University of Galway

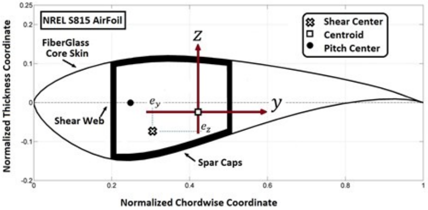
Overview

1. **Applications**
2. Multi-objective optimisation algorithms

Wind turbine engineering

In a wind turbine, what **shape** should the sails be?

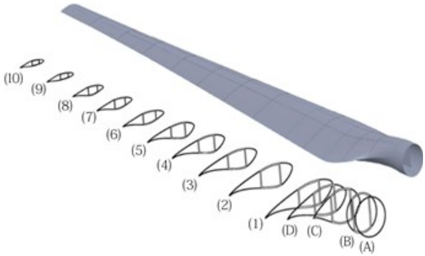


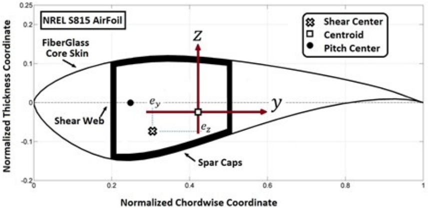


Sheibani and Akbari

Wind turbine engineering

In a wind turbine, what **shape** should the sails be?



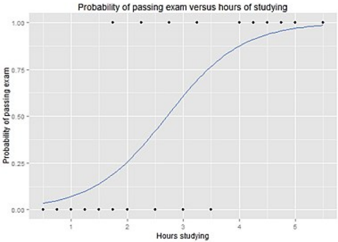


Sheibani and Akbari

We program a finite-element simulation to estimate the **annual energy production** for a given shape. We **maximise**.

We also want to **minimise** construction cost.

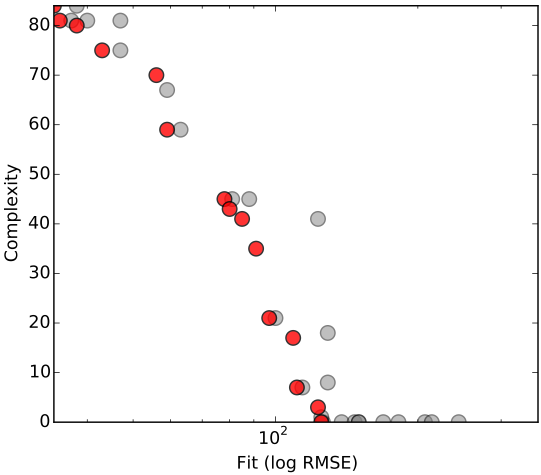
Regression with regularisation



Wiki

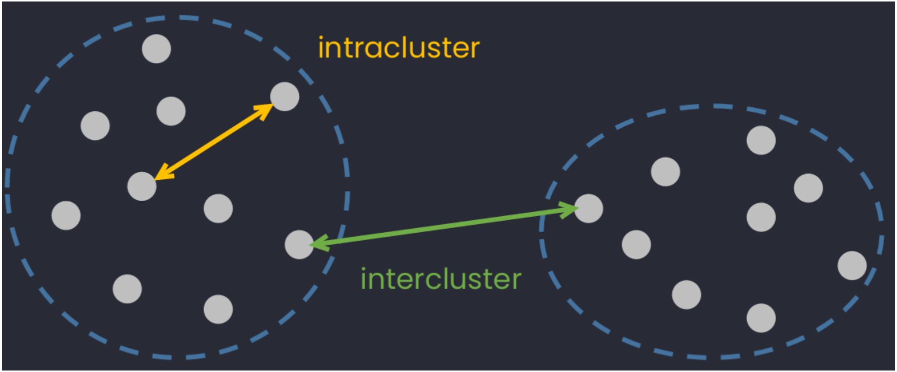
We want to **minimise** the error on the training data, but also **minimise** model complexity, i.e. use **regularisation**.

Genetic Programming symbolic regression



Minimise regression error, minimise equation complexity.

Clustering



https://dinhanhthi.com/metrics-for-clustering/

We want to **minimise** the sum of intra-cluster distances and **maximise** the sum of between-cluster distances.

Print Shop job selection

- Busy print shop, many jobs arrive each morning
- Every job j gives some profit (*value*) v_j, and requires time (*weight*) w_j
- We can only work $w_{\max}$ hours
- Which jobs should we choose to max profit while working less than time limit?
- This is a **knapsack problem**

Multi-objective Unit Commitment

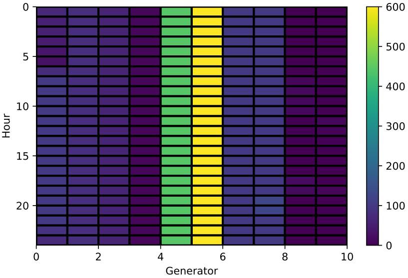
In a small country there are 10 generators of four types: Hydroelectric, Solid fuel, Gas, Solar. Each generator has a lower and upper bound on its production per hour (in MW/h).

	type	LB (MW)	UB (MW)	c (EUR/MW)	p (tons CO2/MW)
A	hydro	10	100	1.4	0.024
B	hydro	10	80	1.4	0.024
C	hydro	10	60	1.4	0.024
D	hydro	1	10	1.4	0.024
E	solid	100	900	4.4	0.82
F	solid	100	600	4.4	0.82
G	solid	10	100	4.4	0.82
H	gas	100	400	9.1	0.49
I	solar	0	70	6.6	0
J	solar	0	20	6.6	0

Minimising cost

Minimise $\sum_{i,j} c_j X_{i,j}$

Subject to meeting demand, X within lower/upper bounds, etc.



- Cost: EUR151518
- Emissions: 23810 tons CO2

Alternative: minimising emissions

Minimise $\sum_{i,j} c_j X_{i,j}$

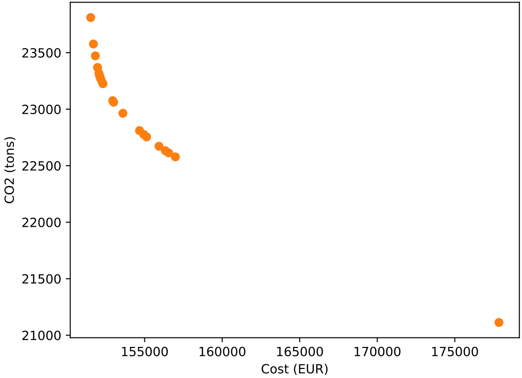
- Cost: EUR151518
- Emissions: 23810 tons CO2

Minimise $\sum_{i,j} p_j X_{i,j}$

- Cost: EUR177846
- Emissions: 21113 tons CO2

A spectrum of solutions

We have considered **only cost**, and then **only emissions**. We really want both! Remember, lower is better for both objectives:



Conflicting objectives

Multi-objective problems arise when we want to maximise or minimise **multiple objectives**

In particular: multiple **conflicting** objectives

Regression

- Maximise R^2
- Minimise RMSE

This are **not conflicting objectives**

We **don't need** multi-objective optimisation.

Overview

Types of algorithms

Weighting schemes:

- Choose a weight for each objective
- Then apply single-objective optimisation

Simple MOO algorithms:

- Pareto archive
- Random-objective tournament selection

Complex MOO algorithms:

- NSGA2 and similar

Weighting schemes

Suppose we have two objectives, maximise f_1 and maximise f_2.

Define some weights w_1 and w_2 such that $w_1 + w_2 = 1$. Define our objective as a weighted sum:

$$f(x) = w_1 f_1(x) + w_2 f_2(x)$$

For any value of w_1, w_2, this is now a **single objective** we can maximise.

Weighting plus grid search

1. For many evenly-spaced values of w_1,
2. Define $w_2 = 1 - w_1$
3. Define the weighted objective and optimise
4. Take the **Pareto front** across the resulting solutions' cost and emissions values.

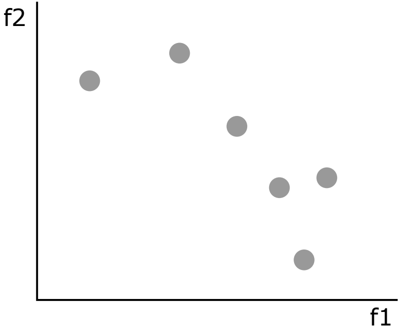
Weighting plus grid search

1. For many evenly-spaced values of w_1,
2. Define $w_2 = 1 - w_1$
3. Define the weighted objective and optimise
4. Take the **Pareto front** across the resulting solutions' cost and emissions values.

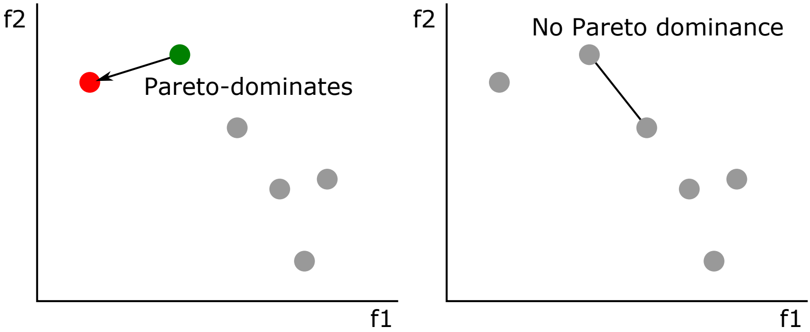
What is the **Pareto front**?

Plotting our options

For any potential solution x, let's calculate $f_1(x)$ and $f_2(x)$. (Suppose higher is better, for both.)



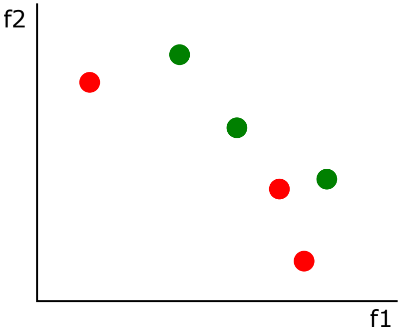
Pareto dominance



We say that an option x_1 **Pareto-dominates** an option x_2 if x_1 is **better than** x_2 on at least one f_i, and **at least equal to** x_2 on all f_i.

An option is **Pareto-dominated** if some other option Pareto-dominates it.

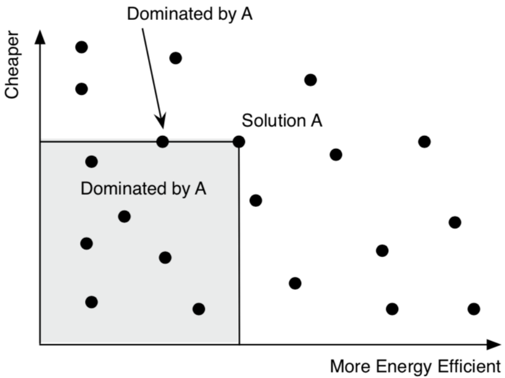
Pareto front



This allows us to discard dominated options, those shown in red.

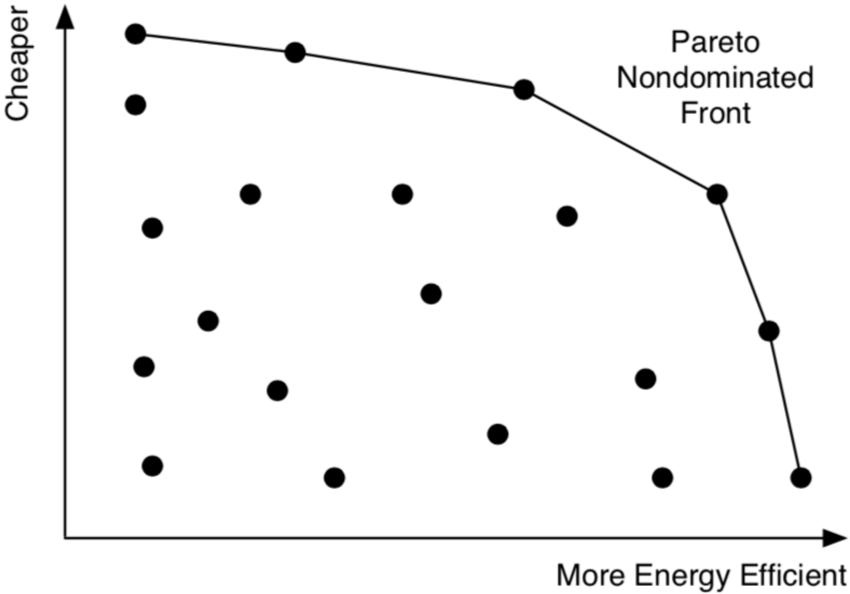
The remainder in green are called the **Pareto front**, the **efficient front**, and similar names. But we have no way to choose between them!

Pareto Dominance

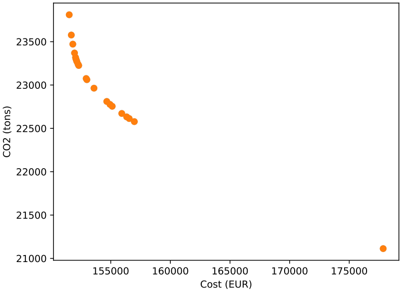


This diagram and later ones are from Luke, Essentials of Metaheuristics, Chapter 7

Pareto Front



Multi-objective Unit Commitment



Remember, all of these solutions are **Pareto-optimal**.

Having obtained this Pareto front, we are finished. A decision-maker now has to **decide**. Sometimes they'll aim for a **knee** in the plot, or an extreme, or they'll introduce new criteria or other issues.

Some simple algorithms

Elitism in MOO

In MOO, the “elite” is just the Pareto front of all points considered so far. It seems natural to **never discard an elite individual**. This leads to a simple algorithm.

Pareto Archive algorithm

1. Create an empty archive, create one individual and add it to the archive.
2. Select random individual(s) from the archive.
3. Generate new individual by mutation and/or crossover.
4. Add the new individual to the archive. If that individual Pareto-dominates any individuals in the set, then **they** are discarded. If it is Pareto-dominated **by** any individual in the set, then **it** is discarded instead.
5. Go to 2.

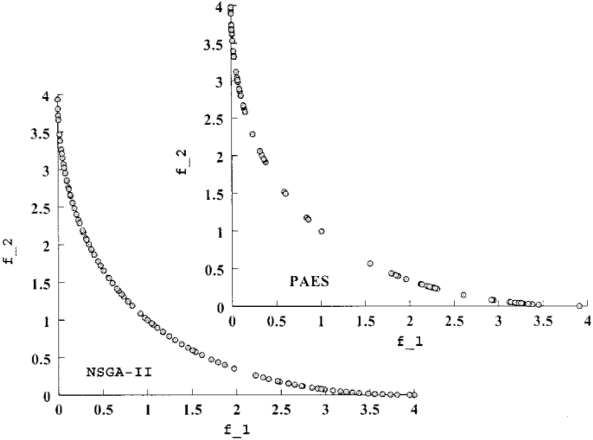
In this algorithm we don't have **generations**. We don't have a fixed population size – in fact the set is likely to grow over time.

Random-objective tournament selection

- This uses a **normal population-based algorithm**, but:
- At each selection event, we choose **one of our objectives at random** and use it to determine the tournament winner
- Variant known as **lexicographic selection algorithm**
- See Luke, Algorithm 95.

Drawbacks

The main drawback of these simpler algorithms (Pareto Archive algorithm and random-objective tournament selection): they fail to fully “spread” the population over the Pareto front.

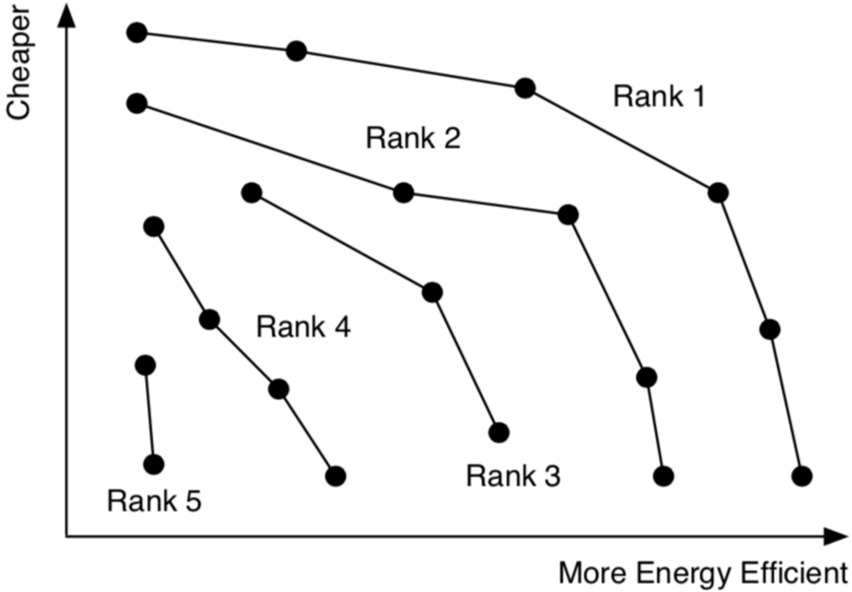


Deb et al.

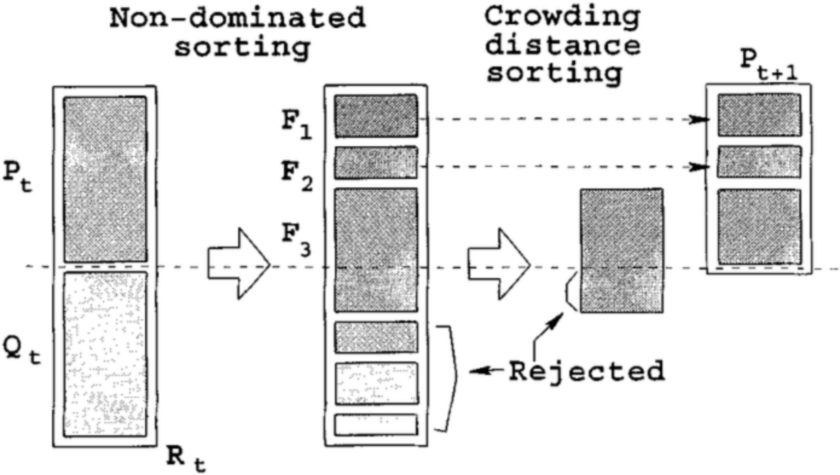
NSGA2

- Non-dominated sorting genetic algorithm 2 (Deb et al., in Blackboard)
- Over 5000 citations.

Sorting


Cheaper
Rank 1
Rank 2
Rank 4
Rank 3
Rank 5
More Energy Efficient

NSGA2 main loop



Deb et al.

NSGA2

Main loop:

- Create children from current population
- Merge with population
- Select according to below procedure
- Discard the rest

NSGA2

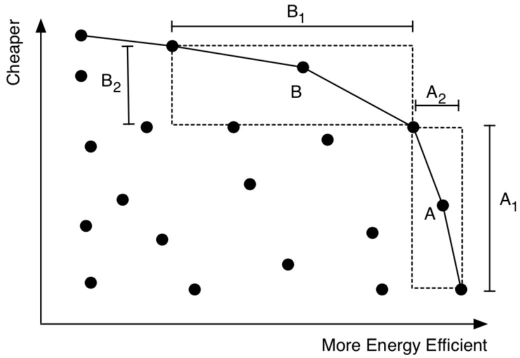
Main loop:

- Create children from current population
- Merge with population
- Select according to below procedure
- Discard the rest

Selection:

- Select all of the Pareto front
- Then remove the Pareto front and re-calculate the next Pareto front, and select all of this
- Repeat (remove, re-calculate, select) until the next Pareto front **would** make our population over-full
- From that Pareto front, just select the most **sparse**.

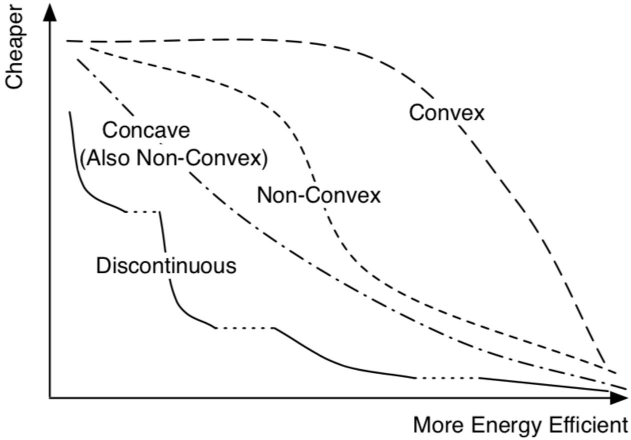
Sparsity



Individual B has sparsity B1+B2. An extreme individual has sparsity ∞. See Luke, Algorithm 102 for details.

The second main idea in NSGA2 is: we would like to achieve a good **spread** of individuals along a wide Pareto front. We select the individuals with highest sparsity, based on how close they are (in objective space) to other individuals in the same front.

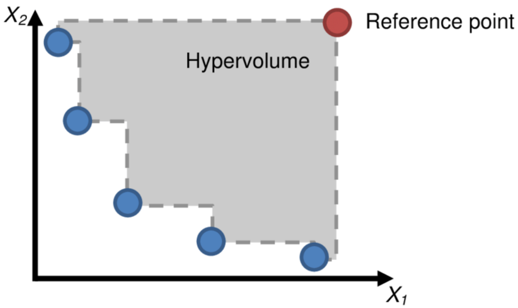
Types of fronts



How good is our solution?

In single-objective optimisation we can just report the best value of the objective. But in MOO, we have to report a whole Pareto front. If we are comparing two algorithms, or judging progress over time, that is tricky. How do we tell whether one Pareto front is better than another?

Hypervolume indicator



Ribeiro

Choose some **reference point** worse than all possible solutions on all objectives and calculate this volume: the larger the better.

DEAP usage

How good is our solution?

Principles:

- Reward Pareto Front for good extreme values
- Reward for diversity across the front (as opposed to clumping)
- If Pareto front PF1 has all the same items as PF2 plus some extra, then PF1 is better.

Recap

- Multiple conflicting objectives
- Pareto dominance
- Basic approaches
  - Weighting schemes, grid search
  - Pareto Archive
  - Random tournament
- **NSGA2**, with selection based on:
  - Ranking of Pareto fronts
  - Sparsity
- Hypervolume indicator to measure algorithm success.

Reading

Luke, **Essentials**, Chapter on Multiobjective optimisation.

Deb, **Multi-objective Optimisation Using Evolutionary Algorithms: An Introduction** (in Bb).

Overview

1. Applications
2. Multi-objective optimisation algorithms