EURO PYTHON PRAGUE & REMOTE

17-23 July 2023

Solving Multi-Objective Constrained Optimisation Problems using Pymoo

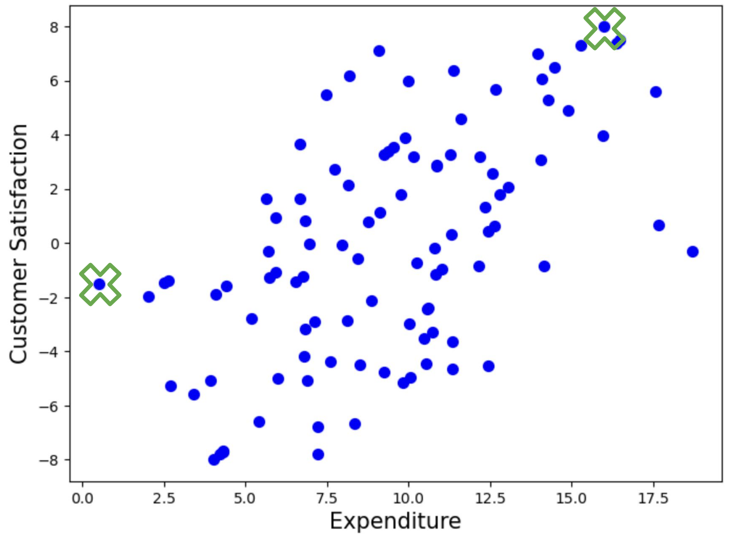
Quick Introduction

- I have had real-life adventures with Optimisation
  - Polymerize : Material discovery and Objective based Experimental Design
  - DataPoem : Maximising ROI for marketing teams
  - Sears : Navigating components for Supply chains
  - Consultancy projects in Manufacturing, FinTech and Energy domains
- Me too! - #Pythonista
- From research to production, at scale
- I still feel like an imposter !

Amazing Experience Last Year!

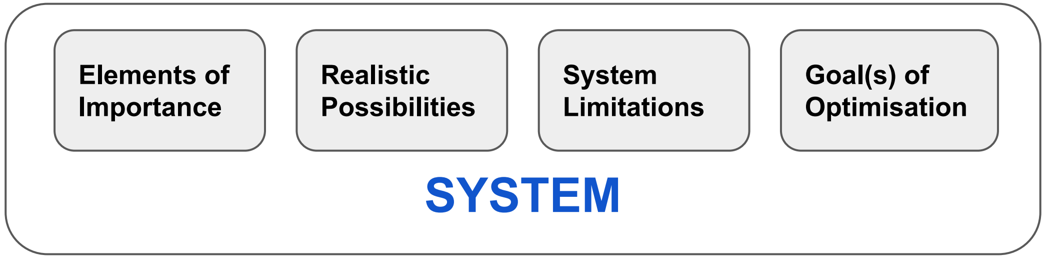
What is Optimisation ?

- age-old problem
  - solving for efficiency
  - peak performance
  - ingenious solutions
- best of all worlds
- Unites us : a core challenge across
- Where there is a ~~will~~ ***trade-off***, there's ~~a way~~ **optimisation**
- Complex and challenging
- almost all cases have more than one optimal solution
- Monopolies are rare!



Here's a not so simple start -
- Maximise Satisfaction
- Minimise Expenditure

Core Components



- Decision Space / Variables
- Bounds
- Constraints
- Objective Function**(s)**

A feasible solution(s) : One where boundaries of individual decision variables are met, system constraints are satisfied and objective(s) are achieved

$$Maximise\ f(x) = 5\alpha + 13\beta^2 - 0.8\gamma$$

where,

$$-12.5 < \alpha,\ \beta < 18\ ,\ \text{and}$$

$$\gamma > 0$$

while,

$$\sqrt{\left(1.8\alpha + 3\gamma - 9\right)} > 45.61$$

Let’s take an example

Supply Chain Optimisation

- Supply must meet demand
- No compromise on a healthy cash flow
- Adjust for delays in logistics
- Cost of transportation
- Product Expiration Deadlines
- A balance of cost and quality
- …

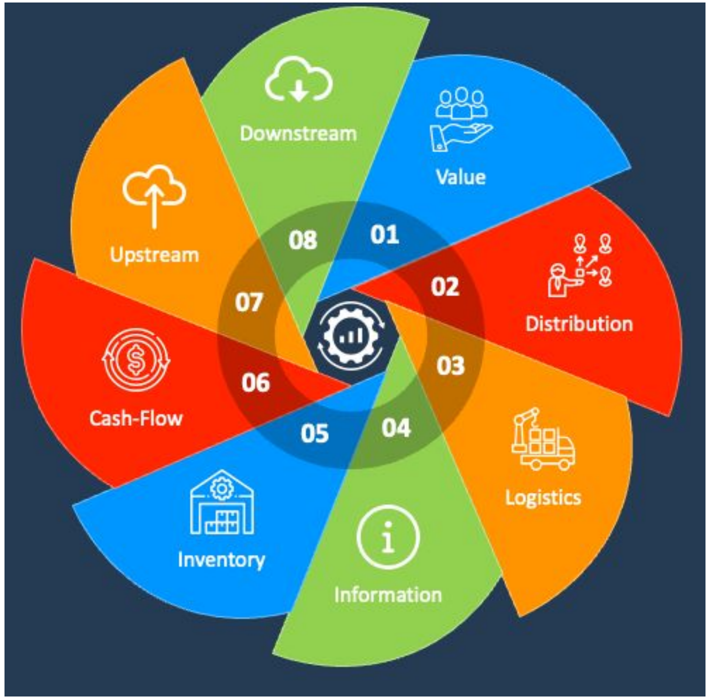
Source : Sight Machine

Finance

- Distribution of funds across items
- Risk to Reward Balance
- Execution time and price change
- Portfolio Management
- ….

Let me Translate

- **Decision Variables**
  - Inventory, In-store , Logistics Cost, Supplier Credit etc.
- **Bounds**
  - 70% < Inventory Capacity < 90%
  - 10% < Supplier Credit < 40% (of order value)
  - ….
- **Constraints**
  - 2M < Net Cash Balance < 4.5M
  - Logistics Capacity Utilisation > 90%
  - ….
- **Objectives**
  - Maximise Operational Profit
  - Maximise Product availability
  - Minimise Delivery Time



Defining an Optimisation Problem

- **Identify Problem Type**
  - Single Objective
  - Multi Objective (2-3)
  - Many Objective (>=4)
  - Selective Priorities
  - **Complexity +** : Objective**s** may not converge, functions may have inflection points, undefined regions in the decision space etc. Plan for a fail safe

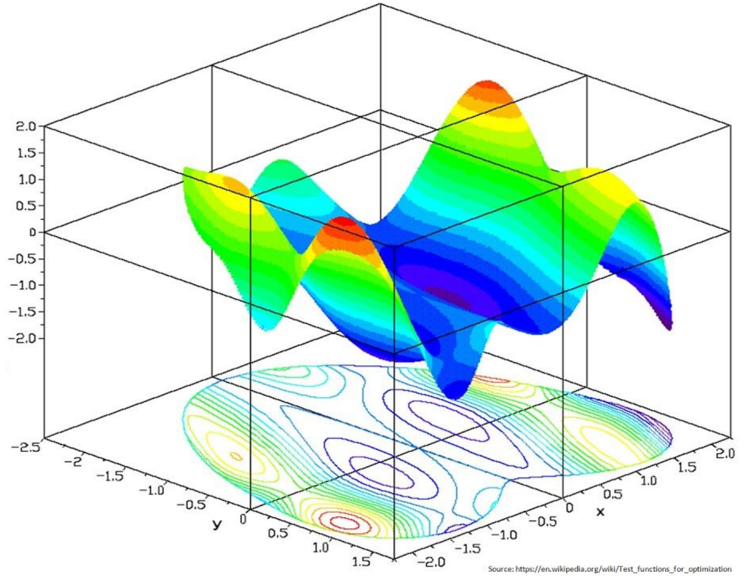
- **Evaluate the Decision Space**
  - Be cautious with decision variables
  - Synchronise domain expertise
  - As much as possible, reduce size
  - **Complexity ++** : Variables may have restricted precision, mixed variable types, mutual exclusion etc

Source : campusq.com

Defining an Optimisation Problem

- **Detect Logical and Systematic Constraints**
  - Sense checks
  - Practicality, relationship inconsistencies
  - Explorative limitations
  - **Complexity ++** : Compliance, Feasibility, Non-deterministic regions in joint decision space etc.

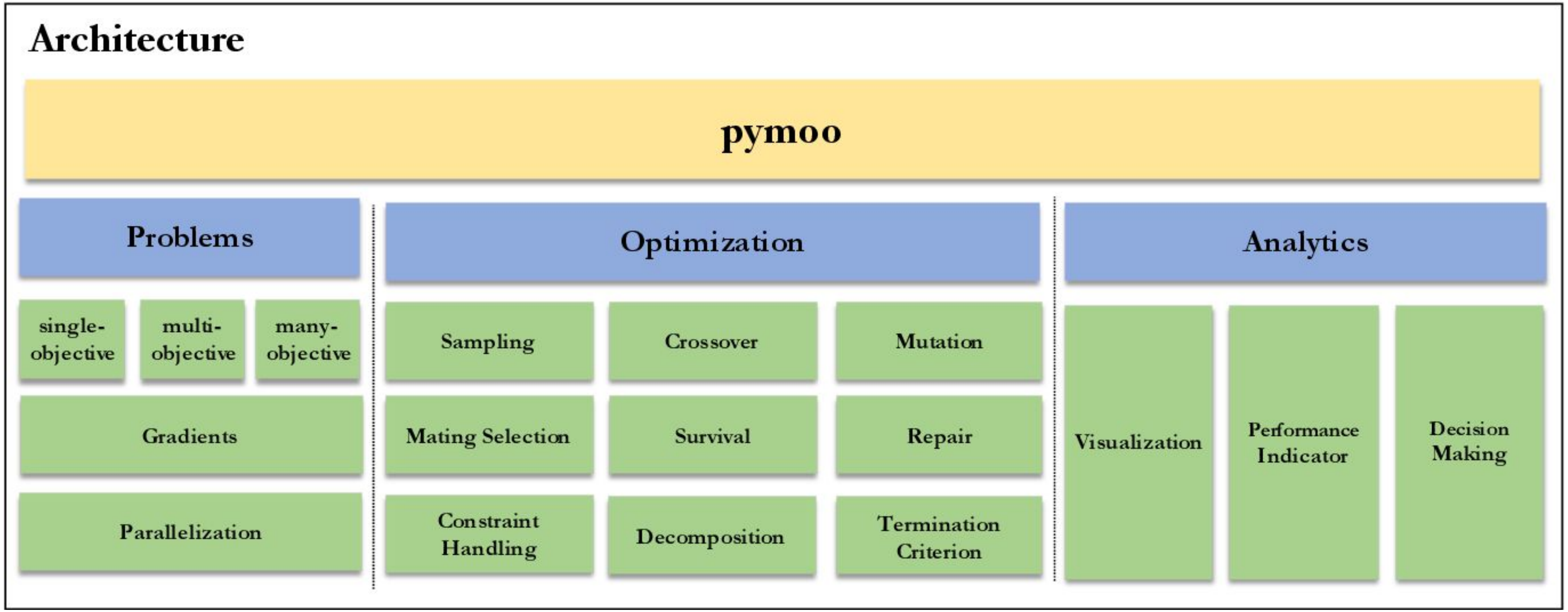
- **Construct the Objective Function**
  - Deterministic, well, if you're lucky!
  - Statistical ? Algorithms, ML, Deep Learning
  - Dynamic & continuous process ? RL
  - **Complexity + :** Dependence on underlying pattern, inconsistency in relationships, dataset sizes, checks and balances for objectives, penalties of varying strengths, high trade-off points



Source: https://en.wikipedia.org/wiki/Test_functions_for_optimization

Pymoo API

State-of-the-Art Modular Python Framework with Object Oriented Interface for Multi Objective Constrained Optimisation with all-round features



Pymoo API

Some noteworthy features of Pymoo -

- A large bouquet of Algorithms for Single, Multi and Many Objective problems
- **Diverse data type support**
  - Custom , Mixed , Binary, Discrete etc.
- **Performance Analytics**
  - Visualisation
  - Convergence
  - Generational Distance
- **Execution**
  - Hyperparameters
  - Custom classes for Mutation, Sampling, Selection, Crossover, Repair etc.
- **Constraint Handling**
  - Penalty
  - Epsilon Handling
- Easy to design solutions with modular and object oriented interface