LabGenius

Protein Design by Multi-Objective Optimisation

Eyal Kazin

@eyalkazin
eyal@labgeni.us

LabGenius

Optimisation

The procedure of finding and comparing ***feasible solutions*** until no better can be found

Decision Space
What we control

Objective Space
What we want

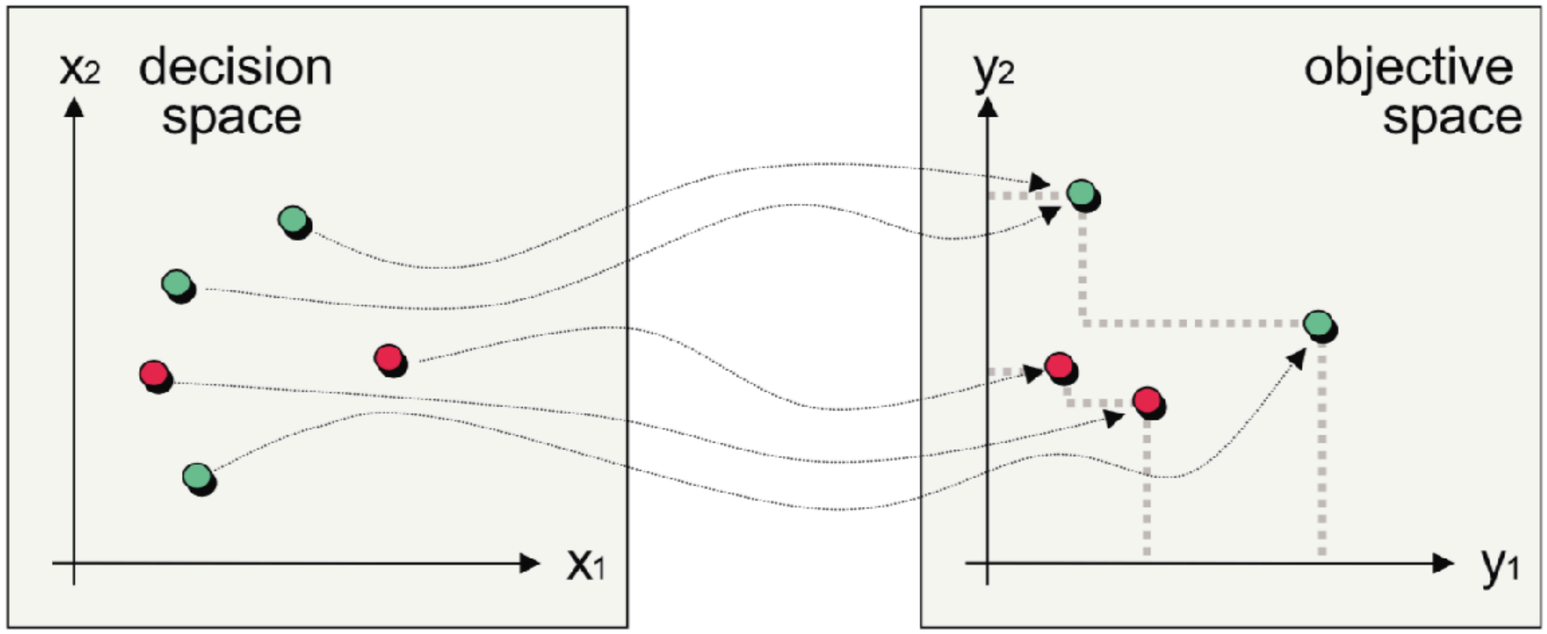


Image: Zitzler et al. (2004)

LabGenius

Multi Objective Optimisation

Optimisation with various objectives is best done simultaneously rather than linearly.

Which option/solution is optimal?

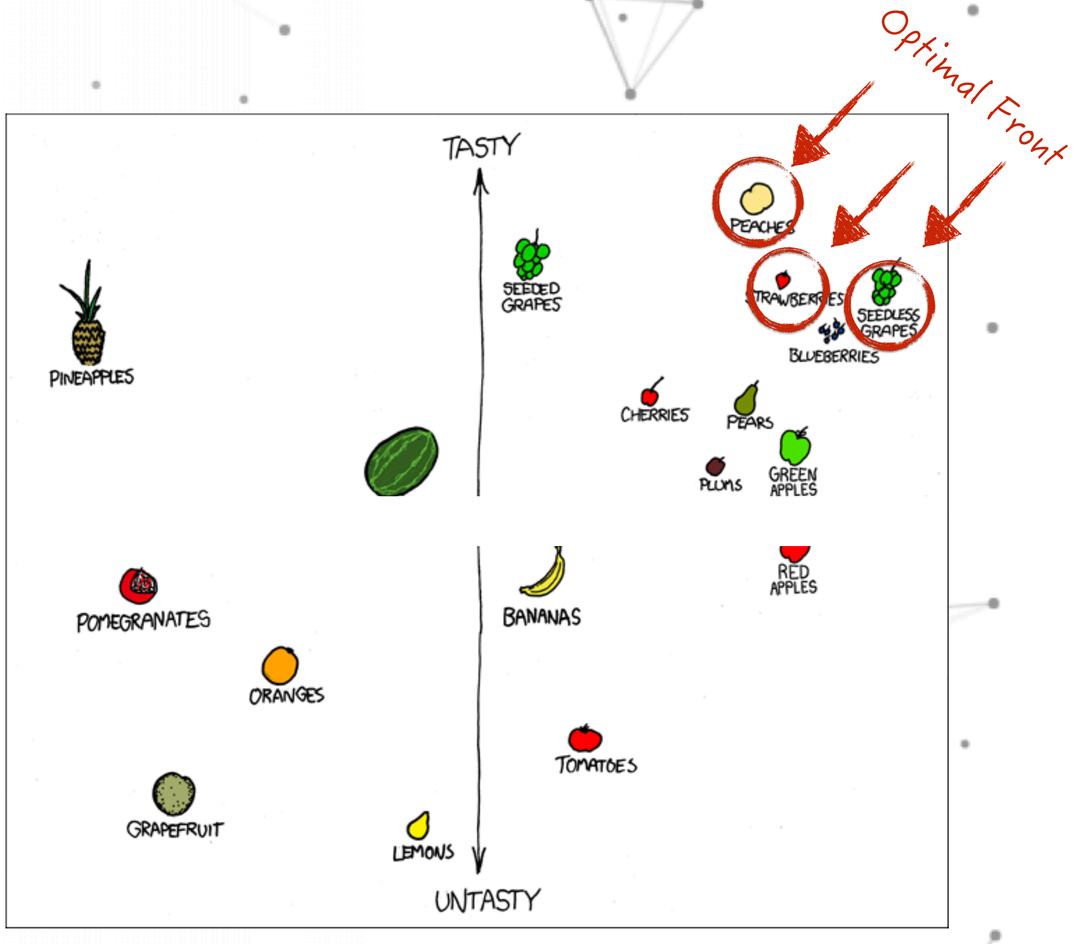


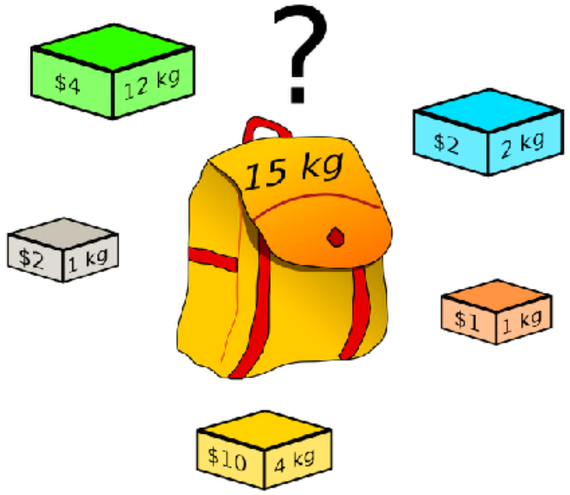
Image credit: XKCD

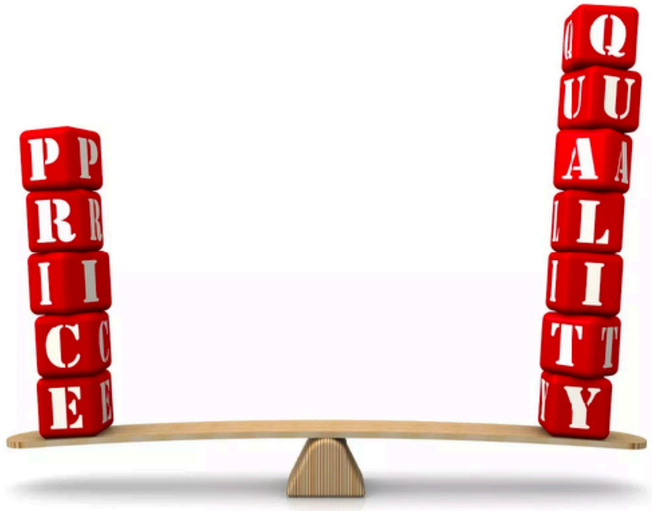
LabGenius

Multi Objective Optimisation

Most real world problems involve more than one objective which conflict

Optimisation for multiple objectives should be done simultaneously, not linearly





LabGenius

Subjective Decision Making For Optimisation

Subjectivity always plays a part in weighing the objectives. But at what stage?

- Decision Before Search—> Single Objective Optimisation

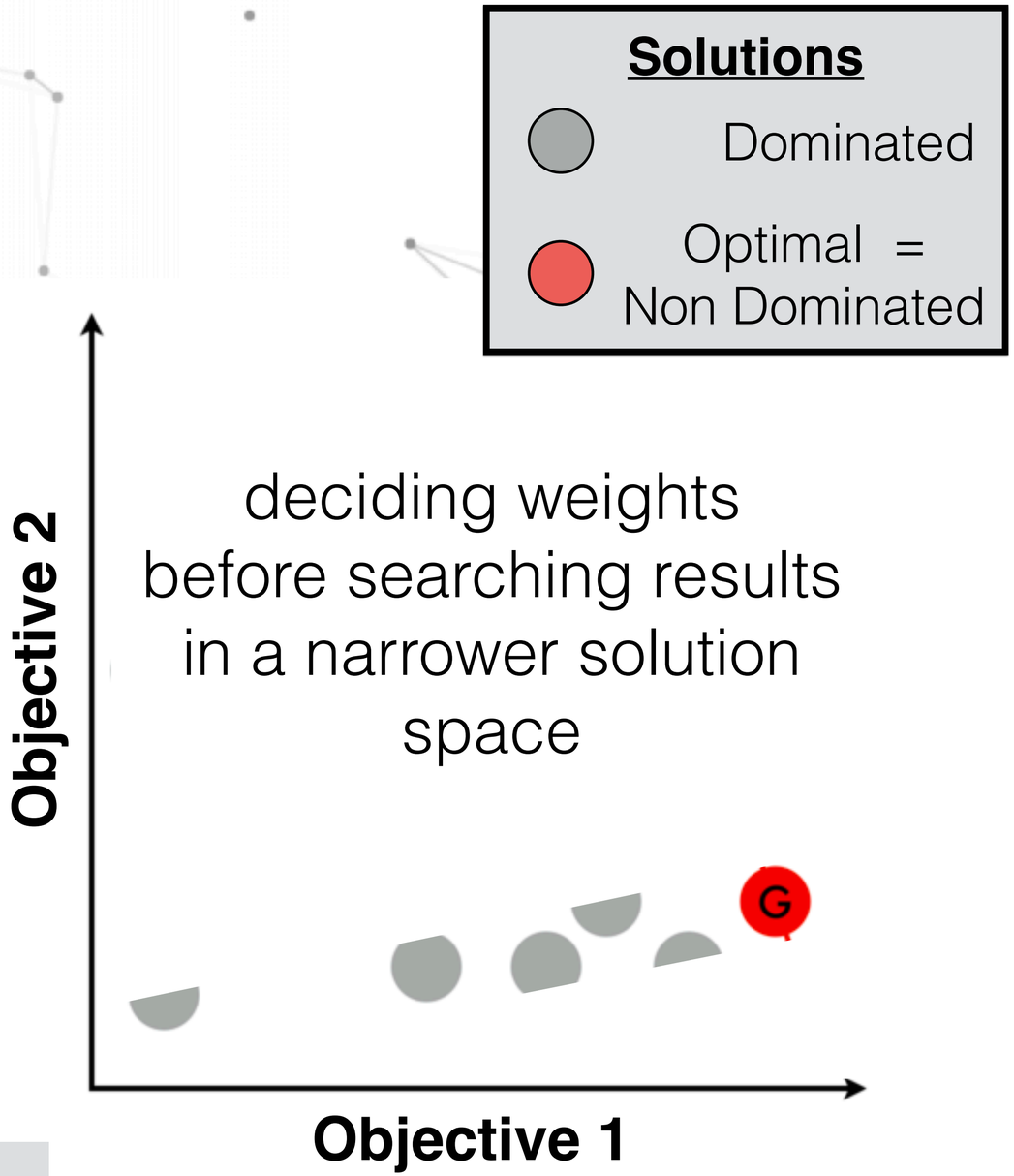
  (e.g, `quality/price`)
- Decision After Search
- Decision During Search

LabGenius

Pareto Front

A set of equally optimal **trade-off solutions**

Optimal solutions may be ranked by a subjective weighting



Note: This approach naturally deals with objectives with different units of measure and scales

Image credit: Wikipedia

LabGenius

Evolutionary Algorithms

- Simple to implement
- Useful for solving multiple conflicting objectives
- Useful for solving intractably large and highly complex search spaces

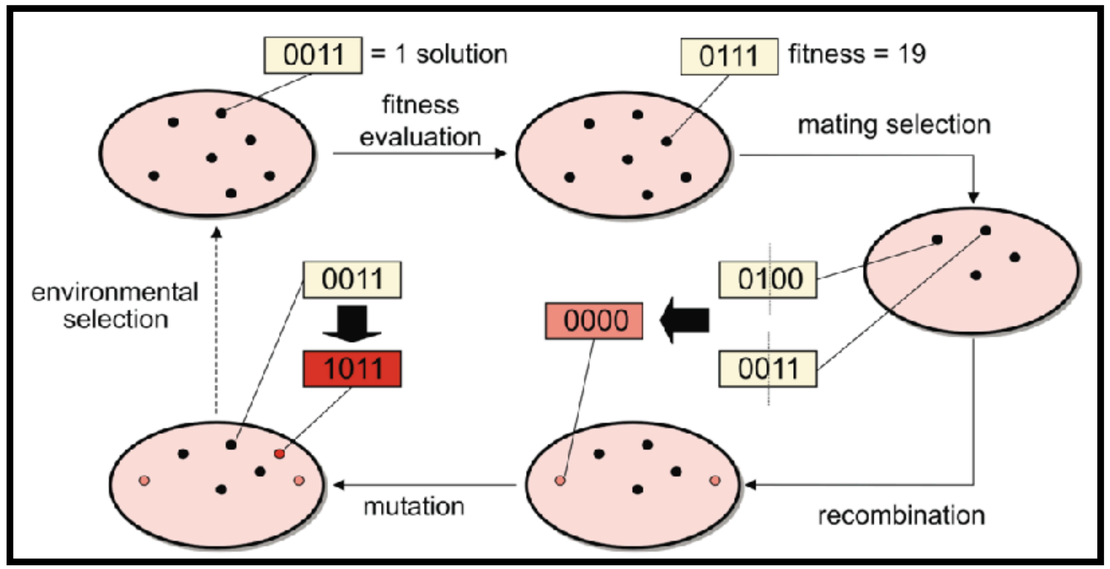


Image: Zitzler et al. (2004)

LabGenius

DISTRIBUTED EVOLUTIONARY ALGORITHMS IN PYTHON

DEAP

Python platform for evolutionary algorithms

Key features:

- Evolutionary algorithms and framework
- Pareto-front algorithms
- Map-reduce for scaling
- As simple as:

```
from deap import tools, creator
```

Cross-over

Mutate

Evaluate

Pareto-Opt

```
                    list,
                    toolbox.individual,
                    seed=None)

toolbox.register("mate",
                    mate,
                    container=creator.Indivi
                    initial_library=initial_
                    )

toolbox.register("mutate",
                    mutate,
                    container=creator.Indivi
                    initial_library=initial_
                    codon_position_probabili
                    keep_zero_probs = keep_z
                    base_unit = base_unit
                    )

toolbox.register("evaluate",
                    evaluate,
                    initial_library=initial_
                    models_lookup=models_loo
                    scores=dict_scores)

toolbox.register("select",
                    tools.selSPEA2)
```

LabGenius **Harness Evolution With AI To Develop Radically New Proteins**

My Day Job @ LabGenius

Empirically improve protein design using Deep Learning and Multi-Objective Optimisation

Decision Space

DNA sequences

```
ATGACTAGACTGGT
CACATGACTGAAAG
TCATAGACGATGAA
CGCCGTAGTGACTT
AATTCTAGACTATC
CTATCGGTGCATCC
```

Objective Space

Protein properties

E.g,

- Stability
- Binding Affinity
- Manufacturability

LabGenius

Contact
@eyalkazin
eyal@labgeni.us

Summary

When should you apply multi-objective optimisaton?

Whenever you have:

- potentially **conflicting objectives**
- **full control** over the decision space

Resources

- Evolutionary Algorithms for Multiobjective Optimisation (Zitzler 1999)
- A Tutorial on Evolutionary Multiobjective Optimisation (Zitzler, Laumanns, Bleuler, 2004)
- python: DEAP
- All animations created with `matplotlib`

```
from matplotlib.animation import FuncAnimation
```

LabGenius

We’re hiring for an experienced Data Engineer!

https://labgenius.workable.com

Contact
@eyalkazin
eyal@labgeni.us

LabGenius

More Slides

LabGenius

Immunogenicity Data used

6,000+ 15-mer peptides derived from the attached peer reviewed articles.

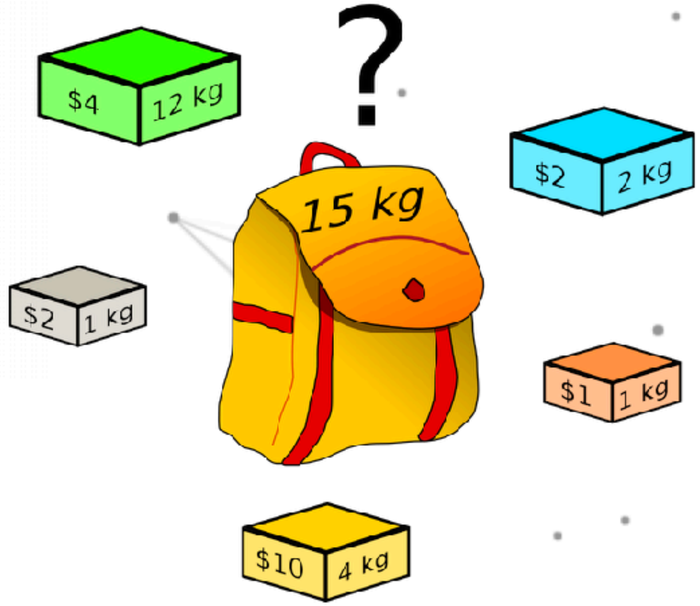
For full details see Dhanda et al. (2018).

(A) Training datasets

Antigen (s)	Peptide selection method	# of donors	Reference	# of epitopes	# of control peptides
Mycobacterium tuberculosis	Overlapping	18	(27)	65	53
	Predicted	28	(28)		1,043
	Overlapping	61	(29)		362
	Confirmed epitopes	61	(29)		137
Timothy grass	Overlapping	25	(14)	60	360
	Predicted	35	(30)		360
	Overlapping	21	(31)		6
	Overlapping	37	(32)		0
House dust mite (HDM)	Overlapping	20	(32)	52	6
Cockroach	Overlapping	19	(33)	71	521
Dengue antigens	Predicted	150	(34)	325	140
Erythropoietin	Overlapping	5	(35)	9	11
CRJ1 and CRJ2	Overlapping	54	(36)	30	18
Mouse allergens	Predicted	22	(37)	82	885

List Source: Dhanda et al. (2018)

Genetic Algorithms



1. **Define *Individuals*** (e.g, set of items)
2. **Define a *Population*** (e.g, a list of Individuals)
3. **Evolving -** In each Generation:
    1. **Mating -** Selecting two or more “parent” Individuals and randomly swapping items to create “offspring” Individuals for a test population
    2. **Mutating -** Mutating selected Individuals (e.g, altering items), to create new Individuals for the test population
4. **Evaluation + Selection -** Evaluate a Fitness score from the objectives (weight, value).

Use the Fitness scores to determine which Individuals survive for the next generation

Image credit: Wikipedia

LabGenius

Decision and Objective Sets

We control the *Decision set* to optimise the *Objective set*

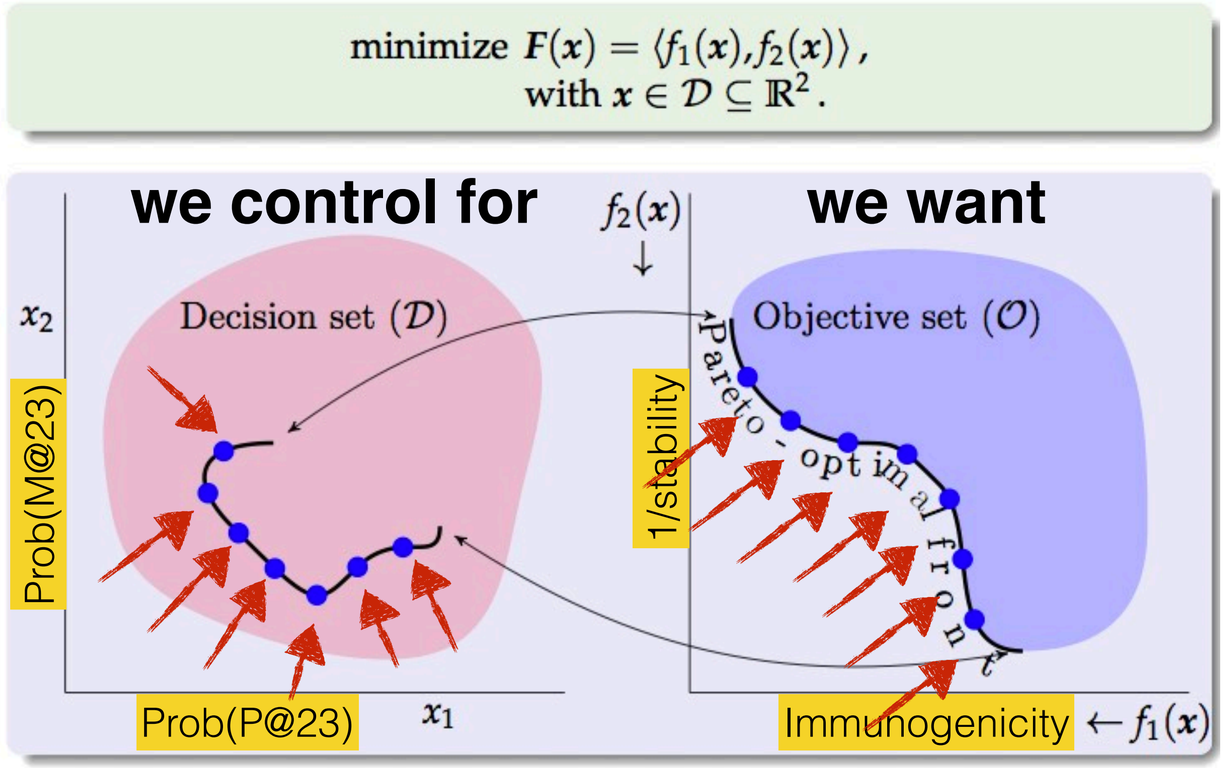


Image credit: Luis Martí