University of Central Lancashire
UCLan

CO3519
Artificial Intelligence

Parameters and objectives
Multicriteria optimization
Terminology and building a glossary

Where opportunity creates success

CO3519 module structure

Upon successful completion of this module, a student will be able to:

1) Explain the theoretical underpinnings of algorithms and techniques specific to artificial intelligence;
2) Critically evaluate the principles and algorithms of artificial intelligence;
3) Analyse and evaluate the theoretical foundations of artificial intelligence and computing;
4) Implement artificial intelligence algorithms.

optimization

agents and decisions

game theory

modelling

knowledge representation

reasoning and learning

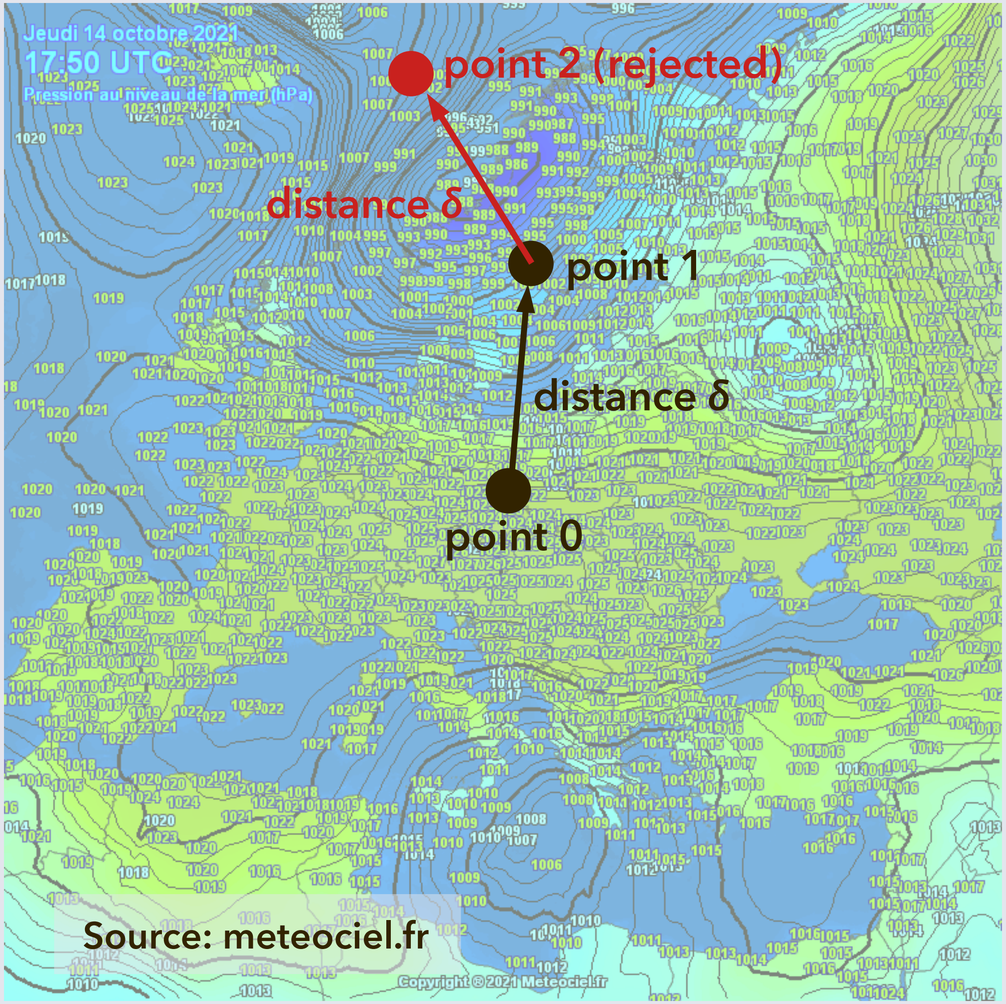
uncertainty quantification

Parameters and objectives

CO3519

26th October 2021

Multivariate optimization (single objective)

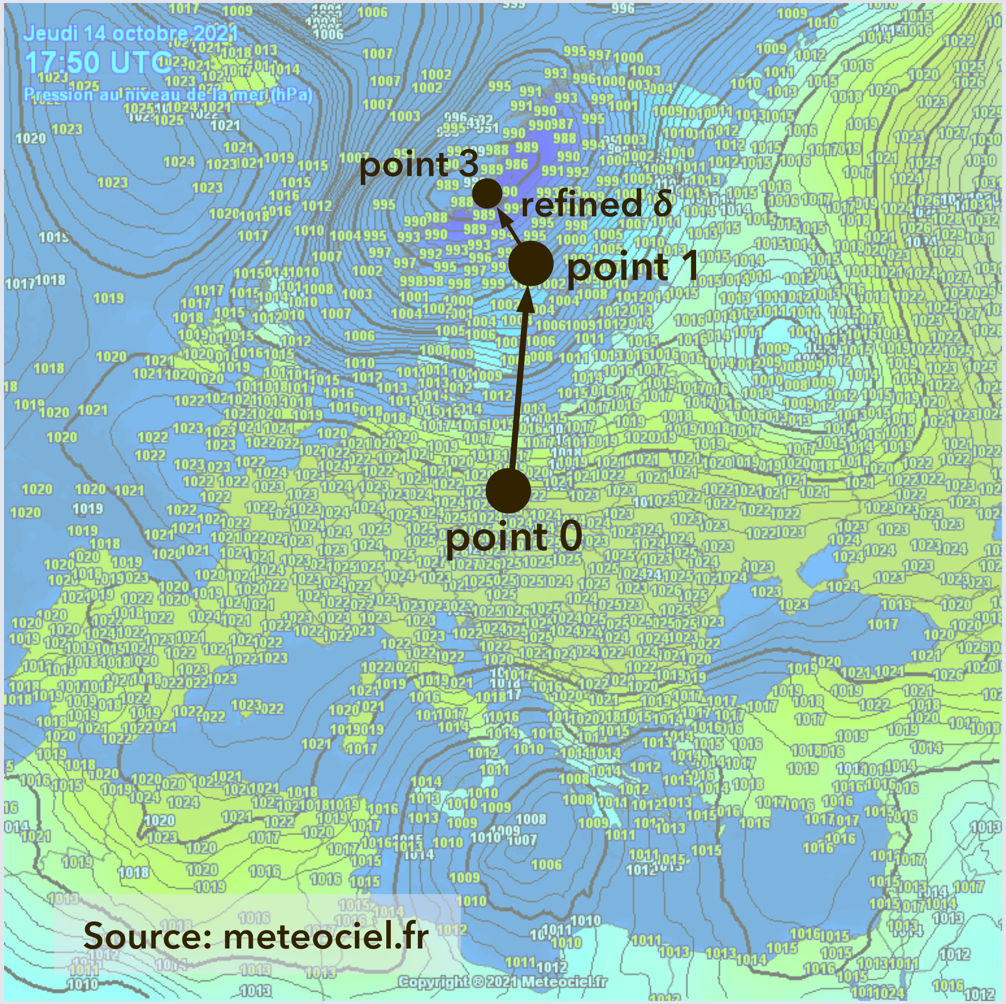


Over multidimensional parameter spaces, minimization by "descent" becomes **steepest descent**, whereas maximization by ascent becomes **steepest ascent**.

In addition to the step size, determined by δ and regulated by α in our codes, a direction needs to be determined.

The direction of steepest descent is **perpendicular to lines of constant cost** (or utility).

Multivariate optimization (single objective)

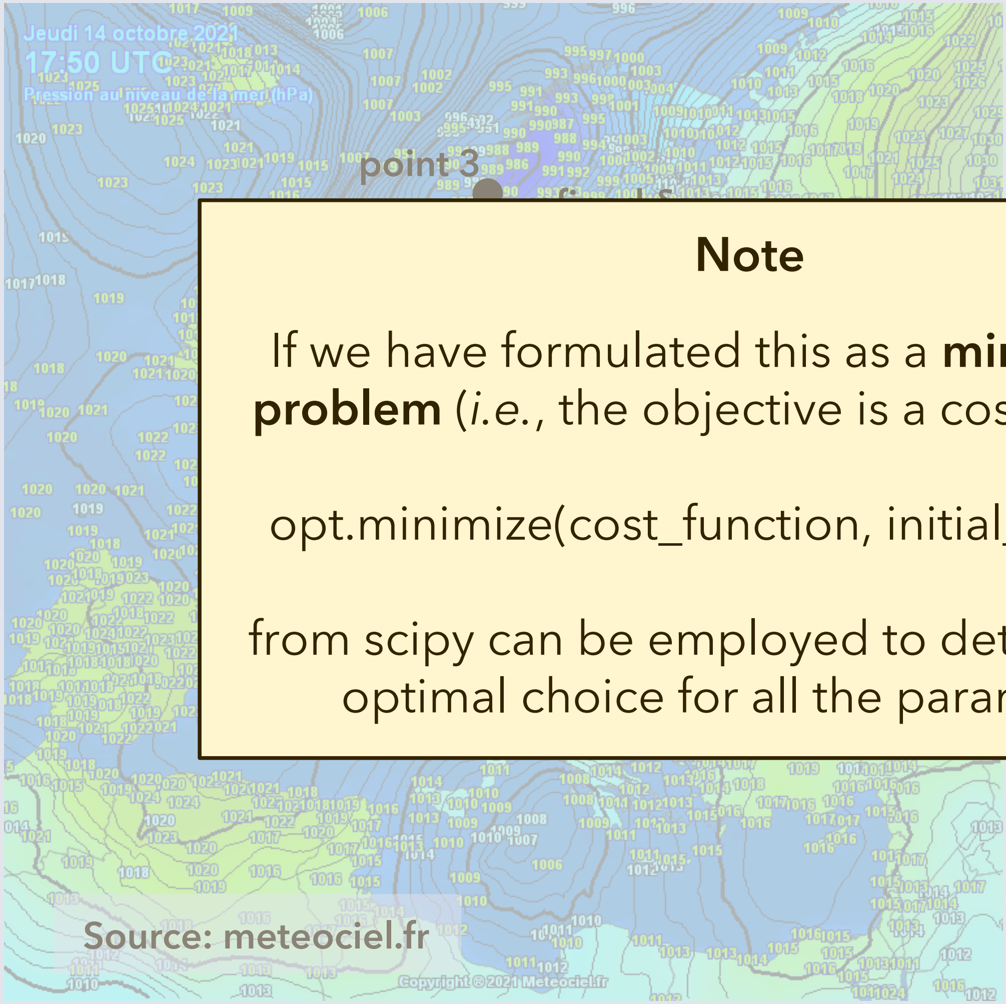


Over multidimensional parameter spaces, minimization by "descent" becomes **steepest descent**, whereas maximization by ascent becomes **steepest ascent**.

In addition to the step size, determined by δ and regulated by α in our codes, a direction needs to be determined.

The direction of steepest descent is **perpendicular to lines of constant cost** (or utility).

Multivariate optimization (single objective)



Over multidimensional parameter spaces, minimization by "descent" becomes **steepest** … ereas maximiza- … nt becomes … **cent**.

… o the step size, … by δ and regula- … our codes, a direc- … o be determined.

The direction of steepest descent is **perpendicular to lines of constant cost** (or utility).

Note

If we have formulated this as a **minimization problem** (*i.e.*, the objective is a cost function),

opt.minimize(cost_function, initial_value, …)

from scipy can be employed to determine the optimal choice for all the parameters.

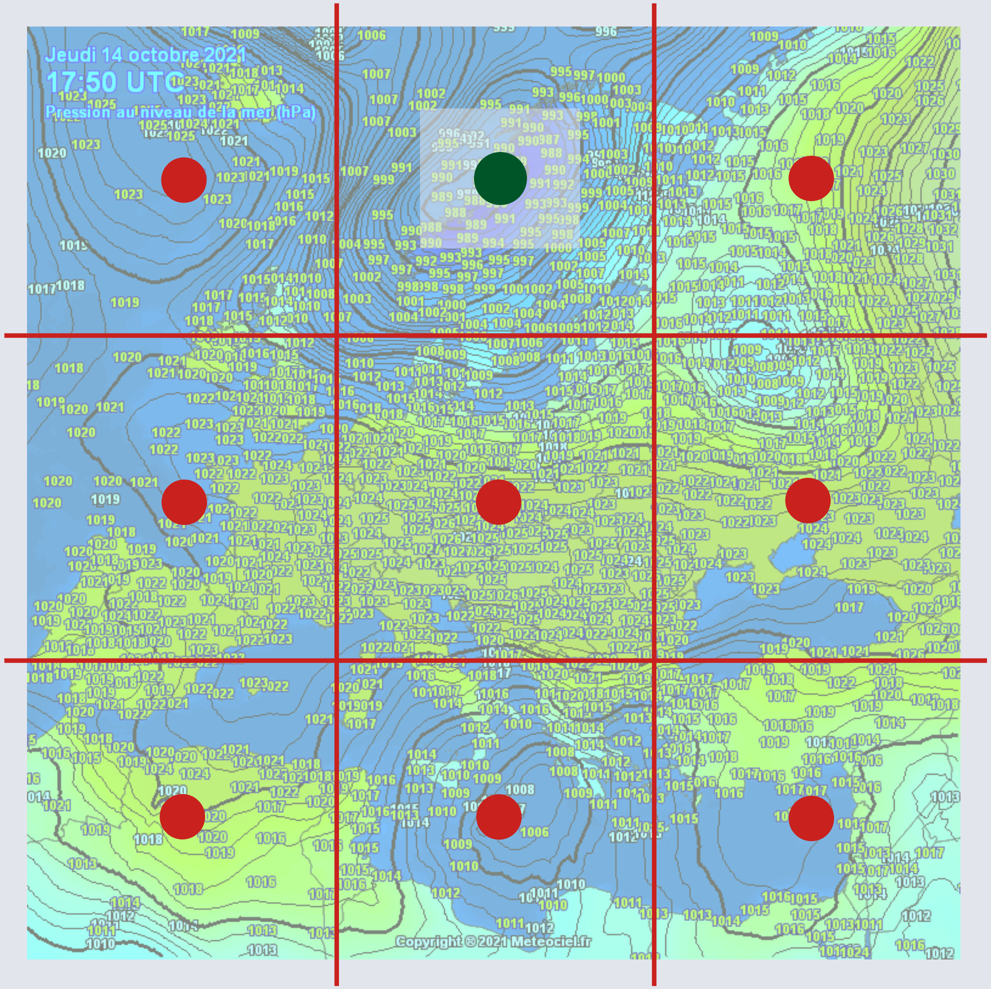
Multivariate optimization (single objective)

Evaluate the objective function at a set of **equidistant points on a**

…

test points for global optimization.

> **Reminder**
>
> Local minimization leads to **a local minimum**, not necessarily to **the global minimum**.
>
> This depends on the **initial value**. For challenging problems, it is advisable to obtain a good initial value first, by applying a **global optimization** scheme.

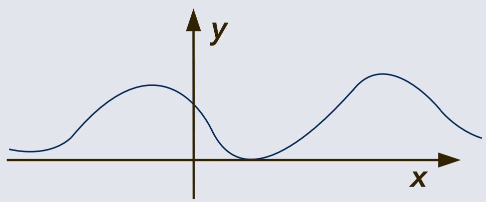


Problems with a single objective

A single maximization objective can be referred to as the **utility (function)**.
If the single objective is to be minimized, it can be called the **cost (function)**.

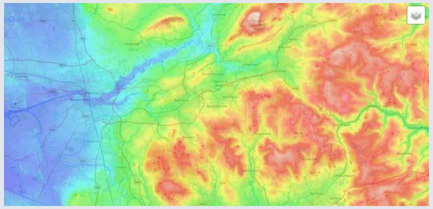
utility or cost

single parameter



done √

multivariate optimization



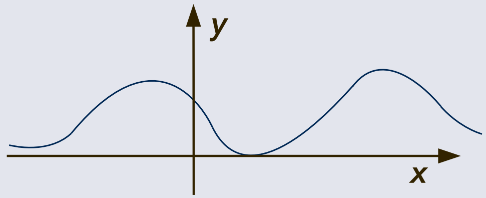
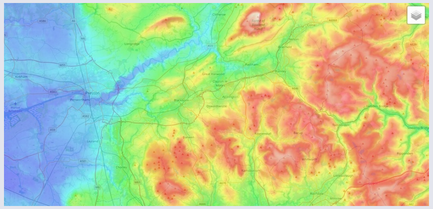
done √

Parameters: Quantites that are part of the solution and **directly in your control**.

Objectives: Quantities that should become as small (or as large) as possible, but can be influenced only **indirectly, through a good choice of parameters**.

Types of optimization problems

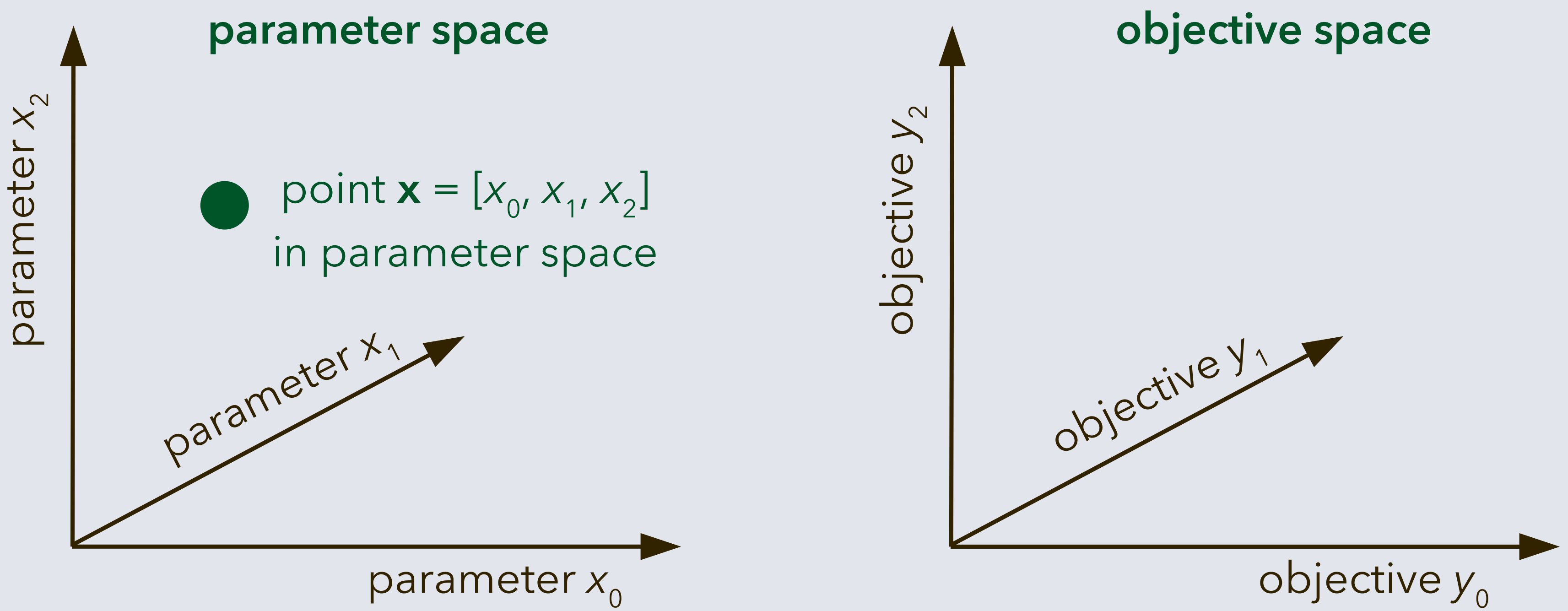
Is there one parameter, or are there multiple parameters?
Is there one objective, or are there multiple objectives?

	utility or cost	multicriteria (MCO)
single parameter		 §
multivariate optimization		

Parameters: Quantites that are part of the solution and **directly in your control**.

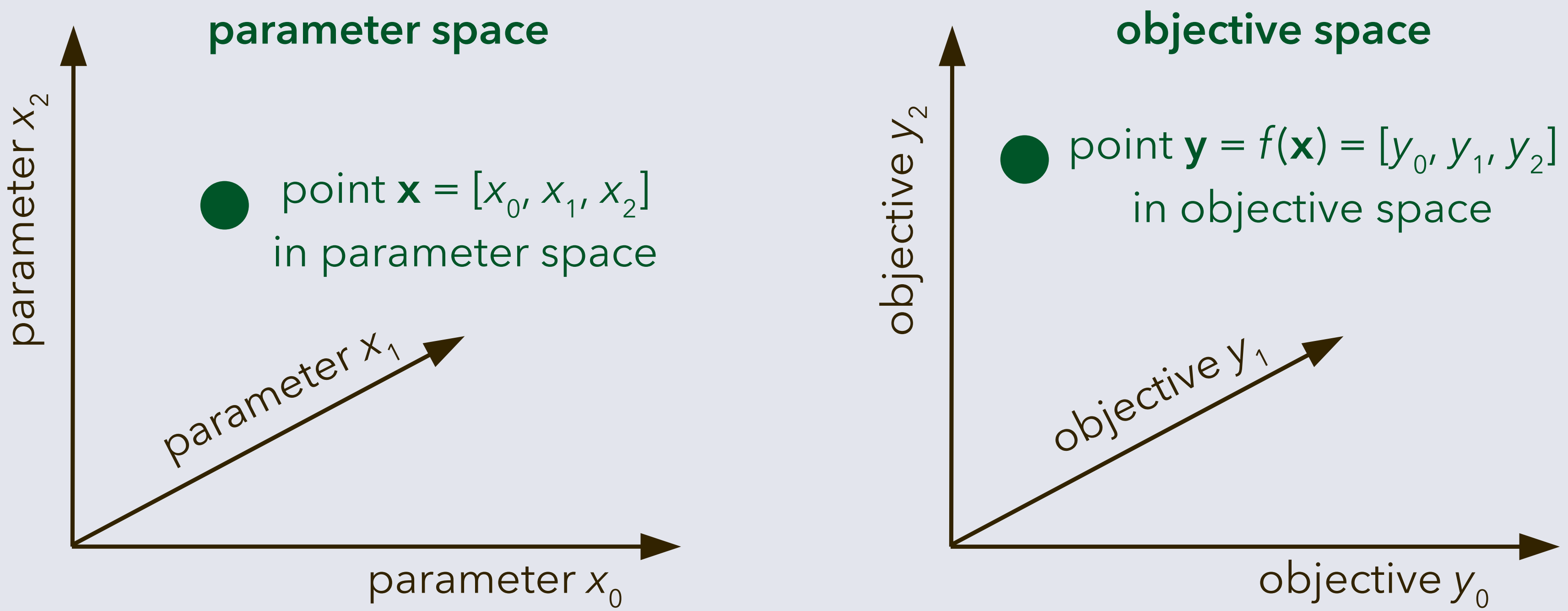
Objectives: Quantities that should become as small (or as large) as possible, but can be influenced only **indirectly, through a good choice of parameters**.

Formalization of optimization problems



In general, there can be any number *m* of parameters (*m*-dimensional parameter space) and any number *n* of objectives (*n*-dimensional objective space).

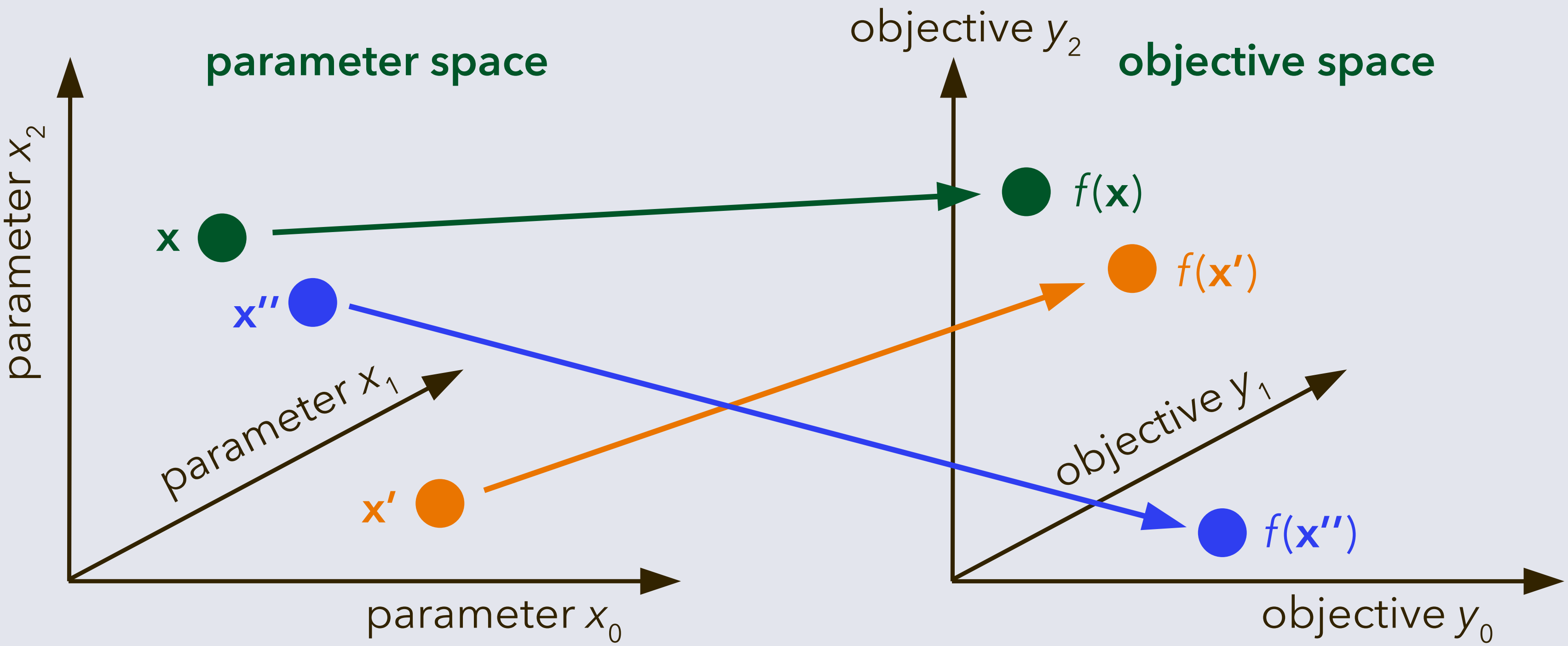
Formalization of optimization problems



In general, there can be any number m of parameters (m-dimensional parameter space) and any number n of objectives (n-dimensional objective space).

The optimization problem is defined by a function f that maps a list of parameters $x = [x_0, \ldots, x_{m-1}]$ to the outcome for the objectives $\mathbf{y} = f(\mathbf{x}) = [y_0, \ldots, y_{n-1}]$.

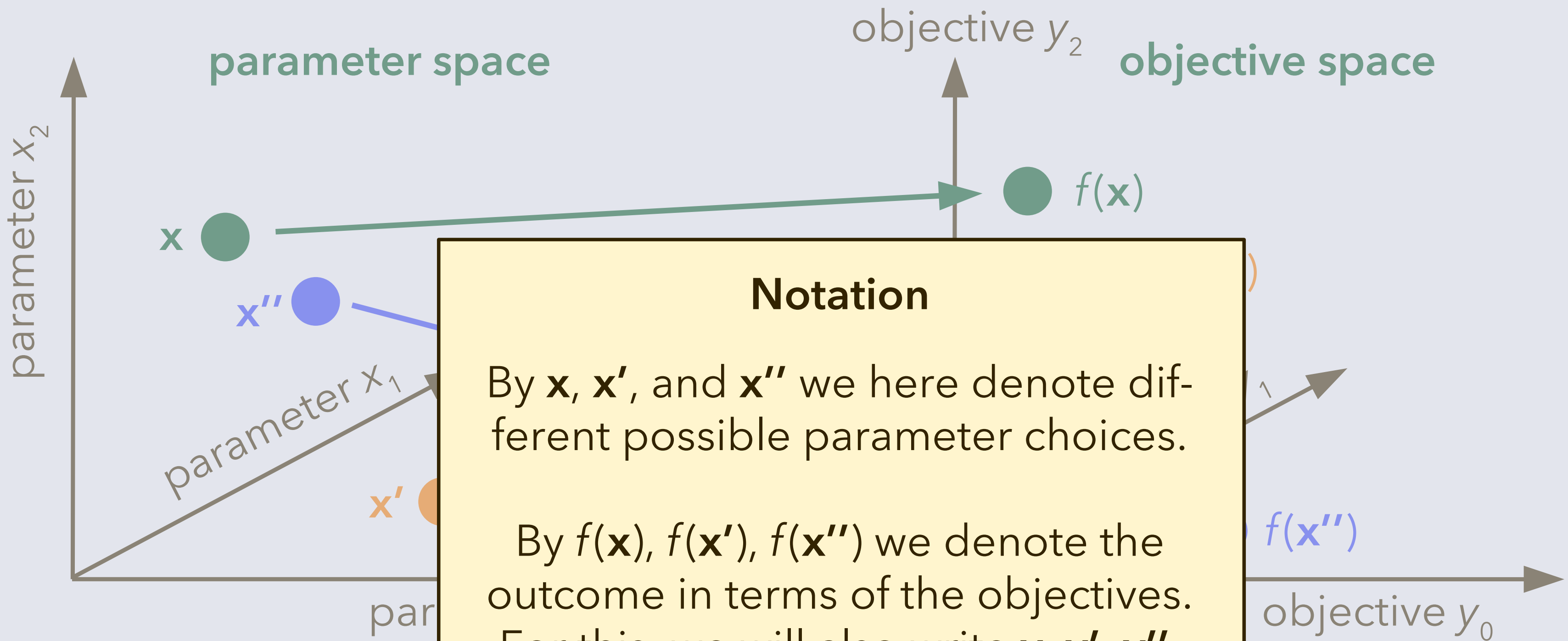
Formalization of optimization problems



Design choices are made by selecting a point in parameter space, but they are **evaluated in objective space**, by comparing the outcomes for $y_0, \ldots, y_{n-1}$.

The optimization problem is defined by a function f that maps a list of parameters x = $[x_0, \ldots, x_{m-1}]$ to the outcome for the objectives $\mathbf{y} = f(\mathbf{x}) = [y_0, \ldots, y_{n-1}]$.

Formalization of optimization problems



Notation

By **x**, **x'**, and **x''** we here denote different possible parameter choices.

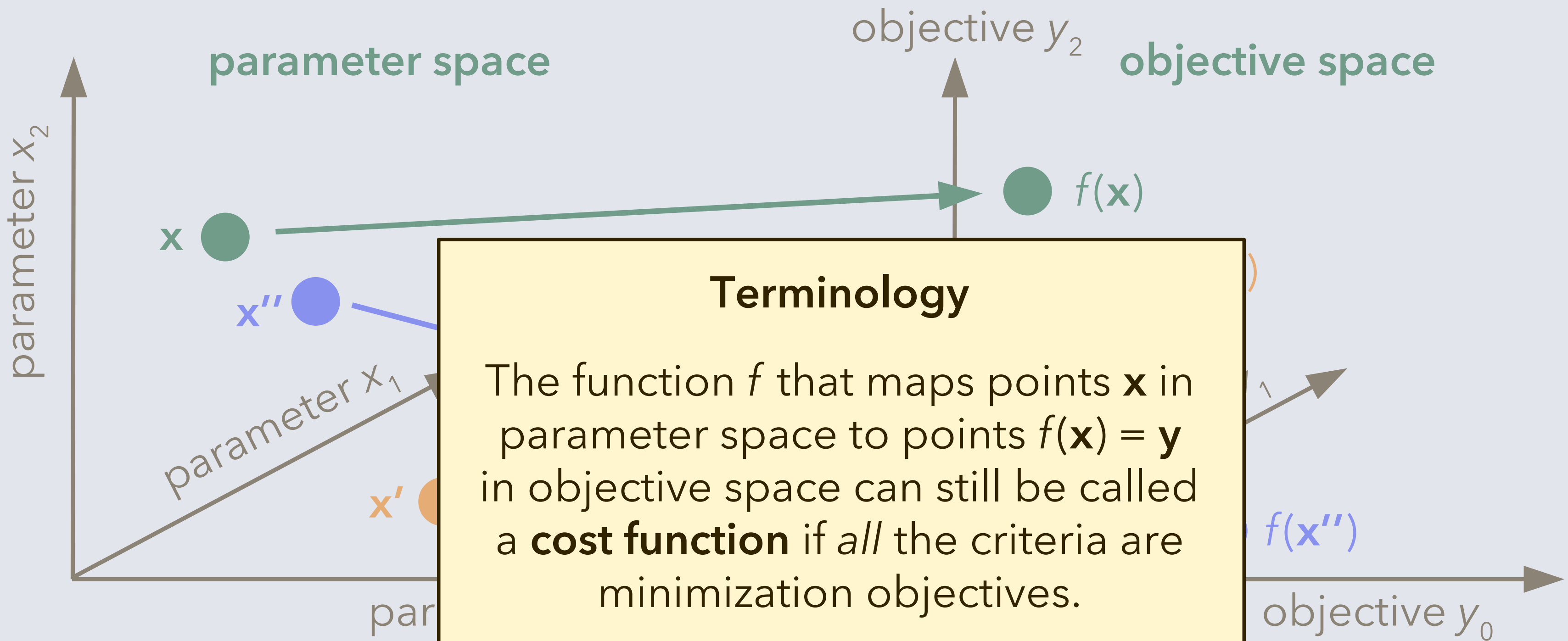
By $f(\mathbf{x})$, $f(\mathbf{x'})$, $f(\mathbf{x''})$ we denote the outcome in terms of the objectives. For this, we will also write **y**, **y'**, **y''**.

(Not to be confused with first and second derivatives. This here are just different possible values.)

Design choices are m… …ace, but they are **evaluated in objectiv**… $y_0, \dots, y_{n-1}$.

The optimization prob… a list of parameters x = $[x_0, \dots, x_{m-1}]$ to the outcome for the objectives $\mathbf{y} = f(\mathbf{x}) = [y_0, \dots, y_{n-1}]$.

Formalization of optimization problems



Terminology

The function f that maps points $\mathbf{x}$ in parameter space to points $f(\mathbf{x}) = \mathbf{y}$ in objective space can still be called a **cost function** if *all* the criteria are minimization objectives.

It is called a **utility function** if *all* the criteria are maximization objectives.

Design choices are m… …ace, but they are **evaluated in objectiv**… … $y_0, \ldots, y_{n-1}$.

The optimization problem is defined by a function f that maps a list of parameters $\mathrm{x} = [x_0, \ldots, x_{m-1}]$ to the outcome for the objectives $\mathbf{y} = f(\mathbf{x}) = [y_0, \ldots, y_{n-1}]$.

Multiple parameters

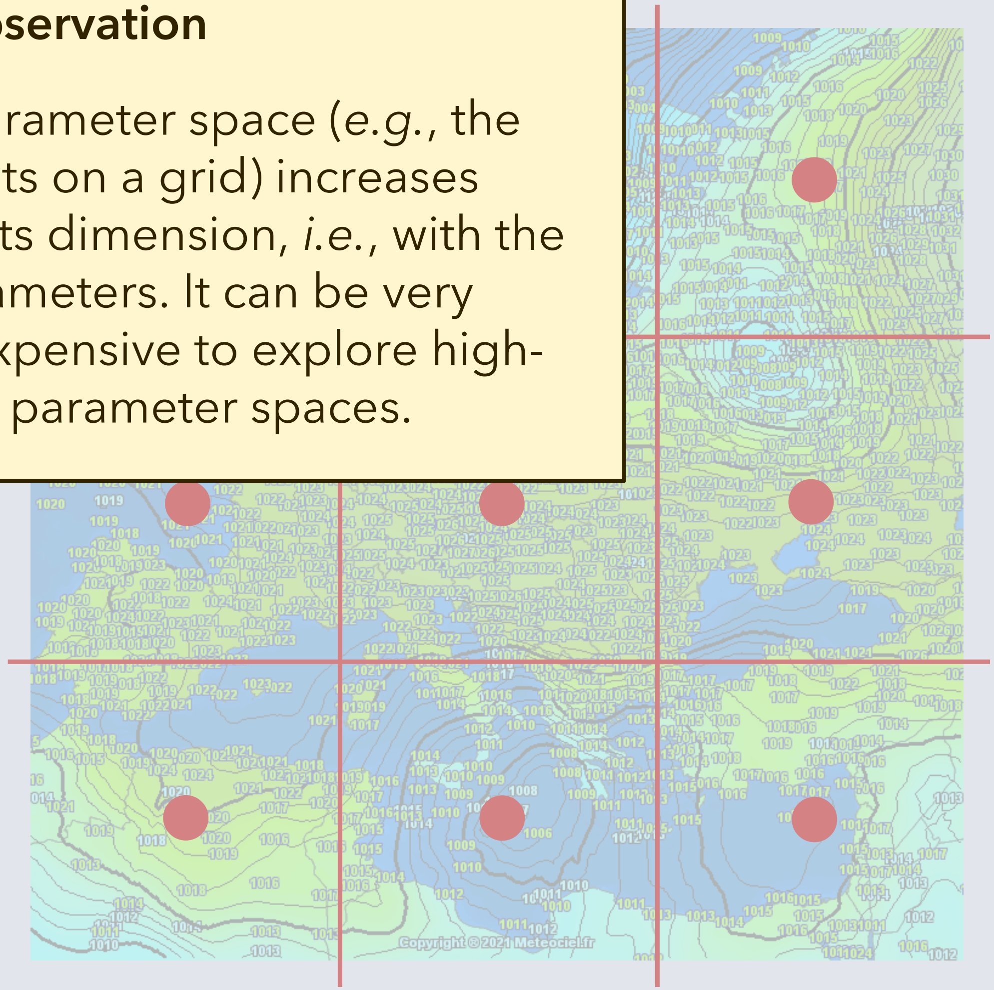
Evaluate the
a set of **equi**
grid, coverin
meter space

Challenge: T
method **expl**
in terms of th
parameters. For *n* parameters with at least two values each, 2^n points need to be evaluated.

In case of high-dimensional parameter spaces, **randomized algorithms** are often used to select test points for global optimization.

Observation

The size of the parameter space (*e.g.*, the number of points on a grid) increases exponentially with its dimension, *i.e.*, with the number of parameters. It can be very computationally expensive to explore high-dimensional parameter spaces.



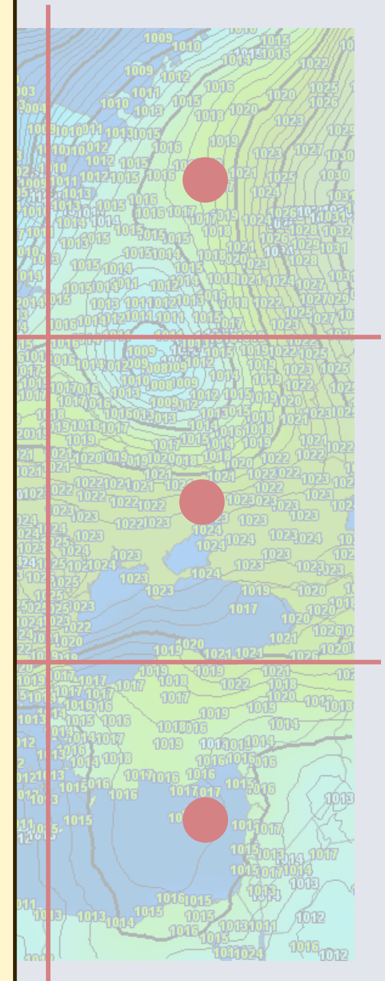
University of Central Lancashire
UCLan

Multiple parameters

Evaluate the
a set of **equi**
grid, coverin
meter space

Challenge: T
method **expl**
in terms of th
parameters.
with at least t
2^n points nee

In case of hig
meter spaces
rithms are of
test points fo



Observation

The size of the parameter space (*e.g.*, the number of points on a grid) increases exponentially with its dimension, *i.e.*, with the number of parameters. It can be very computationally expensive to explore high-dimensional parameter spaces.

It helps to restrict the problem description to the parameters that have the most significant influence on the outcome, neglecting others.

Where it is possible to express the best choice for one parameter x_i as a function of the value of other parameters, $x_i = g(x_j, x_k, \ldots)$, that should be done; x_i can then be discarded.

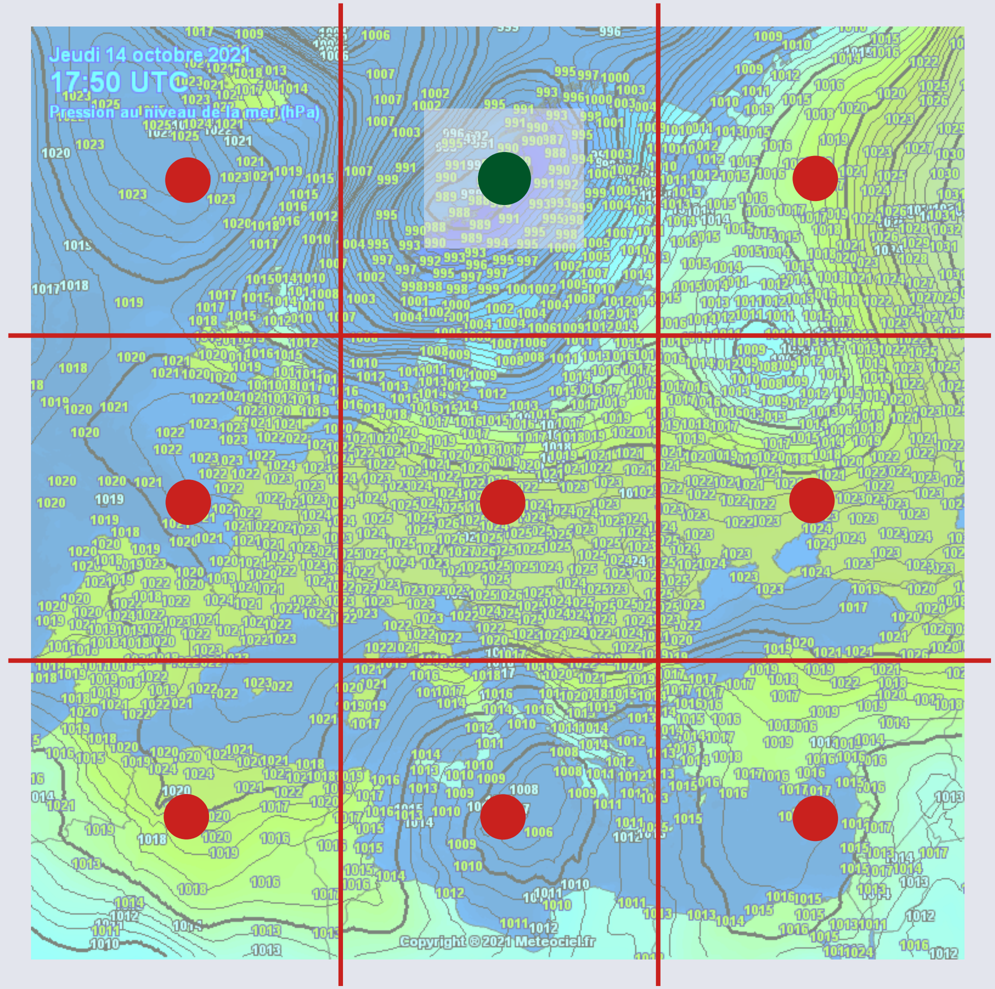
University of Central Lancashire
UCLan

Multiple parameters

Evaluate the objective function at a set of **equidistant points on a grid**, covering the whole parameter space (or the relevant part).

Challenge: The uniform-grid method **explodes exponentially** in terms of the number of parameters. For *n* parameters with at least two values each, 2^n points need to be evaluated.

In case of high-dimensional parameter spaces, **randomized algorithms** are often used to select test points for global optimization.



Multicriteria optimization (MCO)

University of Central Lancashire
UCLan

CO3519

26th October 2021

Multiple objectives

Song by Walther von der Vogelweide (~ 1200)

Diu wolte ich gerne in einen schrîn.
Jâ leider des enmac niht sîn,
daz ***guot*** *und* ***weltlich êre***
und ***gotes hulde*** *mêre*
zesamene in ein herze komen.

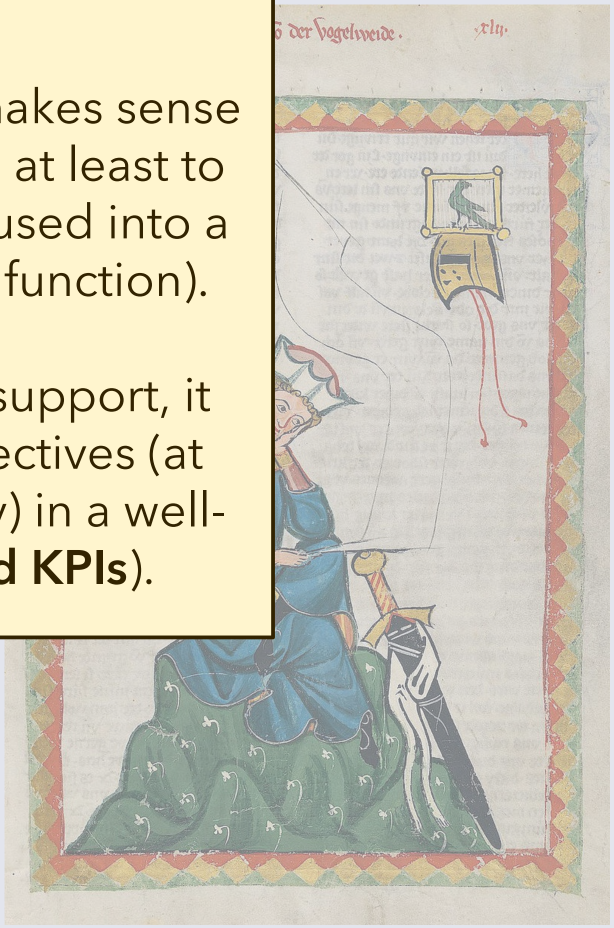
These I would like to have in one box.
But sadly that may not be,
that **goods** and **worldly honour**,
and **God's grace** additionally,
come together in a single heart.

Multiple objectives

Song by W

Diu wolte
Jâ leider d
daz **guot** *u*
und **gotes**
zesamene

These
But sa
that **goods** and **worldly honour**,
and **God's grace** additionally,
come together in a single heart.



Observation

Formulating multiple objectives only makes sense when they are opposed to each other, at least to some extent. Otherwise, they can be fused into a single cost function (or a single utility function).

For use in optimization and decision support, it must be possible to evaluate the objectives (at least qualitatively, better quantitatively) in a well-defined way (**SMART objectives and KPIs**).

Multiple objectives

Song by W

Diu wolte
Jâ leider d
daz **guot** *u*
und **gotes**
zesamene

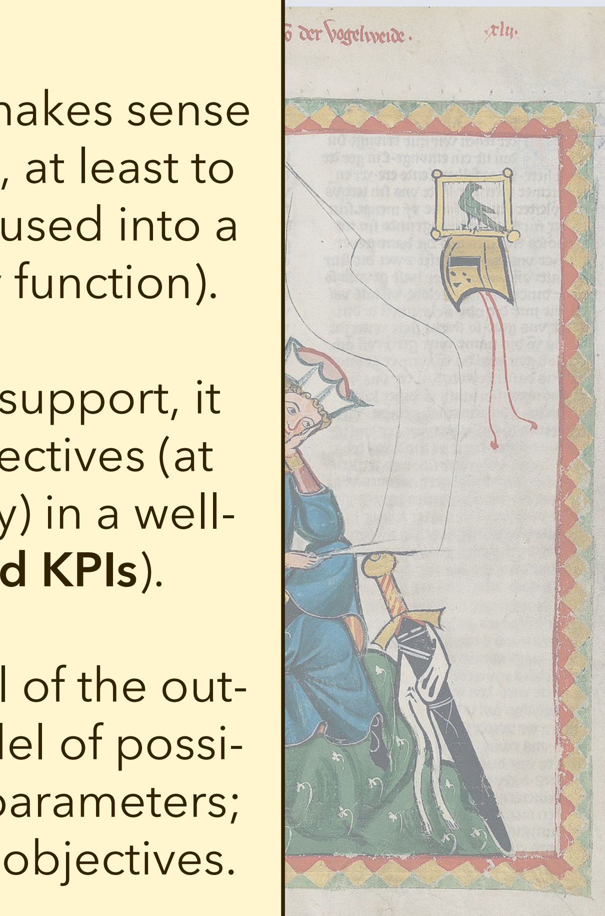
These
But sa
that **g**
and **G**
come t

Observation

Formulating multiple objectives only makes sense when they are opposed to each other, at least to some extent. Otherwise, they can be fused into a single cost function (or a single utility function).

For use in optimization and decision support, it must be possible to evaluate the objectives (at least qualitatively, better quantitatively) in a well-defined way (**SMART objectives and KPIs**).

This often requires **models**: a) a model of the outcome, in terms of objectives; b) a model of possible courses of action, expressible by parameters; c) a function that maps parameters to objectives.



Multiple objectives: Walther’s example

List of SMART objectives

1) My income will increase by at least 10% per year over each of the coming five years.

2) I will be invited to perform for the Bishop of Passau at least once a year over each of the coming five years.

3) I will give at least 10% of my income to the poor in each of the coming five years.

The list of SMART objectives is not directly very suitable for optimization. Why not?

Multiple objectives: Walther's example

List of key performance indicators (KPIs)

1) Annual increase of income, to be determined on an annual basis; target: all ≥ 10%.

2) Invited gigs at Passau, per year (five values); target: all ≥ 1.

3) Share of income given to the poor, to be determined each year; target: all ≥ 10%.

The above are three quantifiable and measurable objectives x_0, x_1, x_2 (or fifteen, $x_0, \ldots, x_{14}$, if the yearly outcomes are considered separately).

Multiple objectives: Walther's example

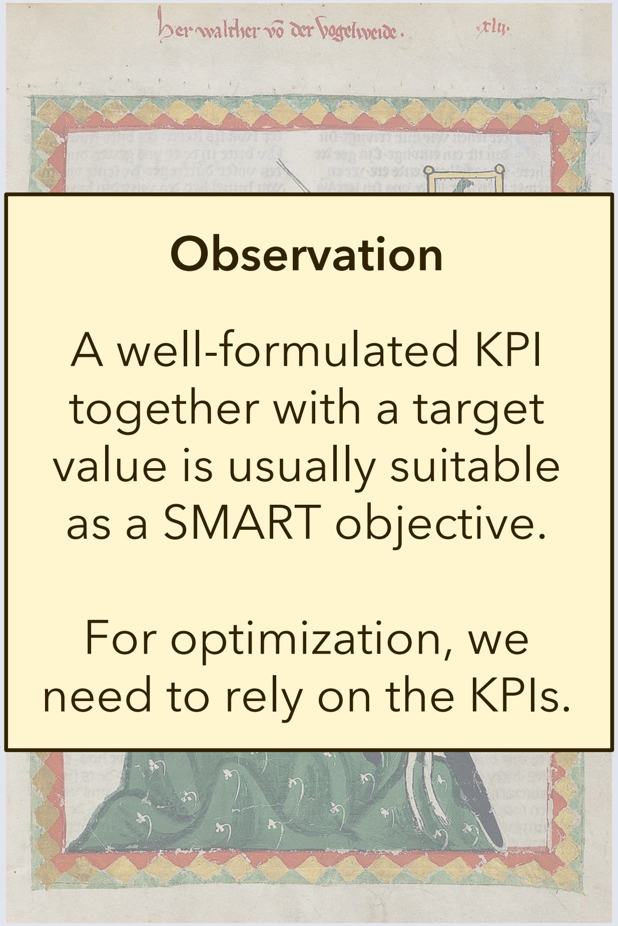
List of key performance indicators (KPIs)

1) Annual increase of income, to be determined on an annual basis; target: all ≥ 10%.

2) Invited gigs at Passau, per year (five values); target: all ≥ 1.

3) Share of income given to the poor, to be determined each year; target: all ≥ 10%.

The above are three quantifiable and measurable objectives x_0, x_1, x_2 (or fifteen, x_0, ..., x_{14}, if the yearly outcomes are considered separately).



Observation

A well-formulated KPI together with a target value is usually suitable as a SMART objective.

For optimization, we need to rely on the KPIs.

Multiple objectives: Walther's example

List of key performance indicators (KPIs)

1) Annual *decrease* of income, to be determined on an annual basis; target: all ≤ –10%.

2) Invited gigs at Passau, per year (five values), multiplied by the *factor –1*; target: all ≤ –1.

3) Share of income *not given to the poor*, to be determined each year; target: all ≤ 90%.

The above are now given as three minimization objectives x_0, x_1, x_2 (or fifteen, x_0, ..., x_{14}, if the yearly outcomes are considered separately).

Multiple objectives: Walther's example

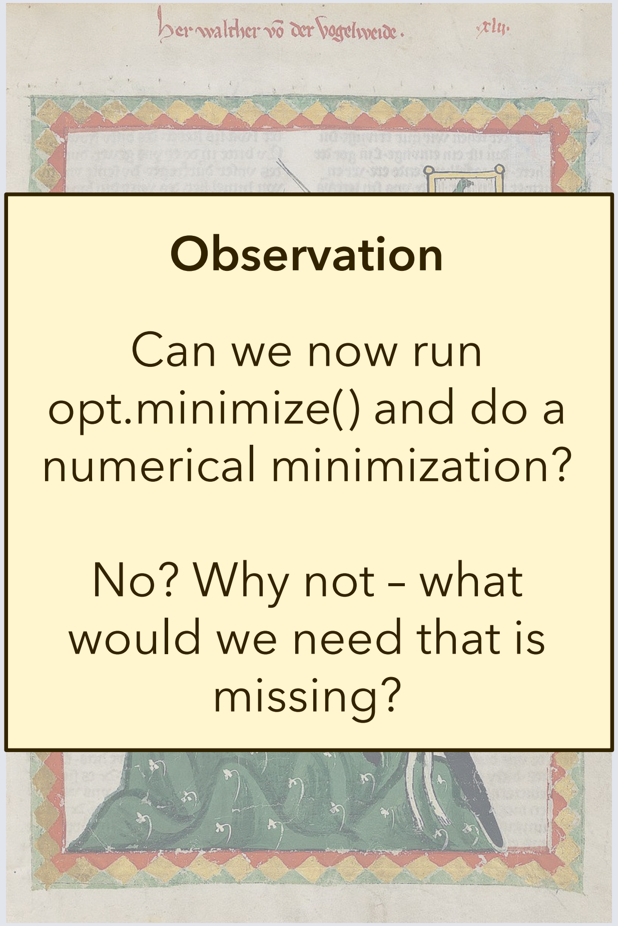
List of key performance indicators (KPIs)

1) Annual *decrease* of income, to be determined on an annual basis; target: all ≤ –10%.

2) Invited gigs at Passau, per year (five values), multiplied by the *factor –1*; target: all ≤ –1.

3) Share of income *not given to the poor*, to be determined each year; target: all ≤ 90%.

The above are now given as three minimization objectives x_0, x_1, x_2 (or fifteen, x_0, ..., x_{14}, if the yearly outcomes are considered separately).



Observation

Can we now run opt.minimize() and do a numerical minimization?

No? Why not – what would we need that is missing?

Parameters & objectives: Investment example

1.2.3. *Cost functions with more than two parameters*

The `multivar` notebook defines a cost function for a hypothetical industrial operation; at planning and design stage, you have direct control over the following parameters:

- The investment i, done a single time, in units of £.
- The amount of goods p to be produced, in units of £/year.
- The depreciation period d (how long it is meant to operate), in units of years.

In the investment-decision example from multivar, we specified three parameters i, p, and d. However, in the underlying model, the amount of goods p that can be produced *without external manufacturers* is given by a function $p = g(i)$ of the value i. This might be simplified to a two-parameter problem.

Parameters & objectives: Investment example

1.2.3. Cost functions with more than two parameters

The `multivar` notebook defines a cost function for a hypothetical industrial operation; at planning and design stage, you have direct control over the following parameters:

- The investment i, done a single time, in units of £.
- The amount of goods p to be produced, in units of £/year.
- The depreciation period d (how long it is meant to operate), in units of years.

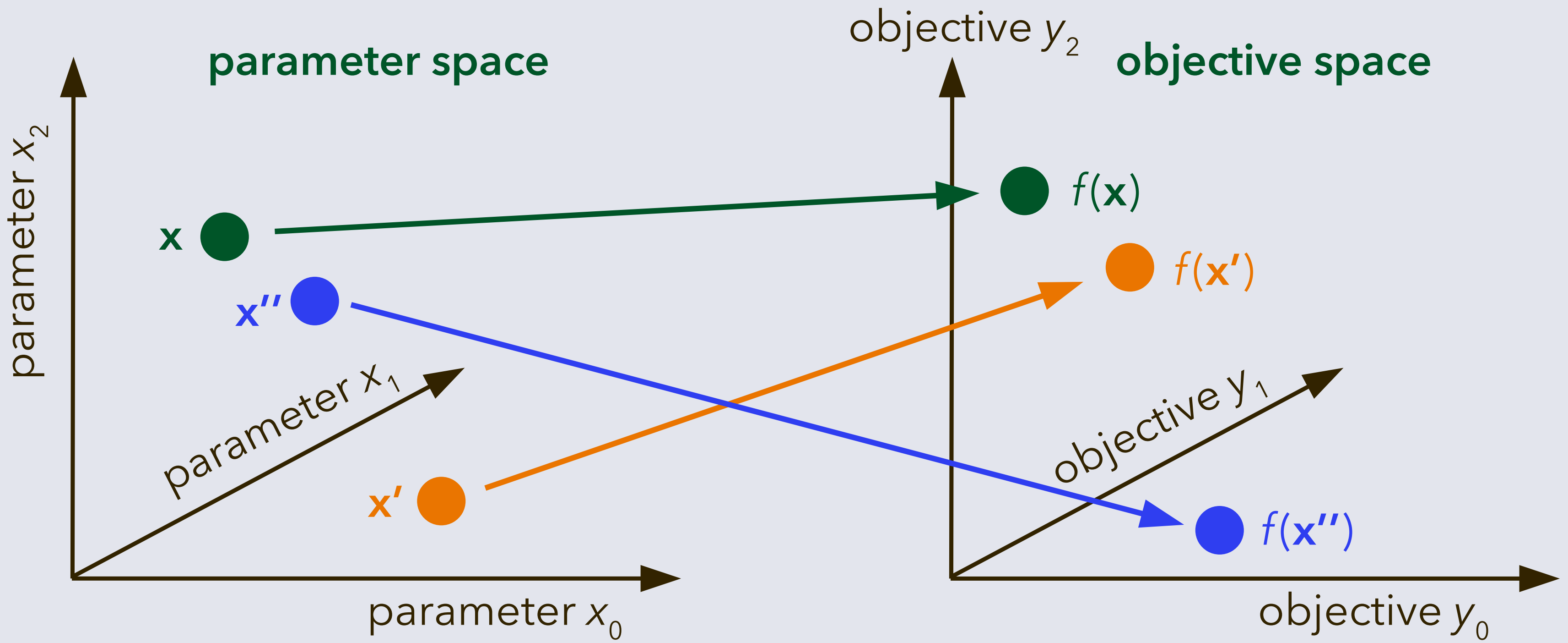
```
total investment         i = GBP 100000
production volume        p = GBP 1000000 per year
depreciation period      d = 7 years

operating cost: GBP 102915.03 per year
depreciation:   GBP 14285.71 per year
prod.cost:      GBP 900000.0 per year
sales contrib.: GBP -1000000.0 per year
==
total deficit:  GBP 17200.74 per year
```

A single-objective cost function was given. However, the evaluation specifies multiple contributions to it.

It may make sense to distinguish two objectives: Minimize proper costs, and maximize sales income.

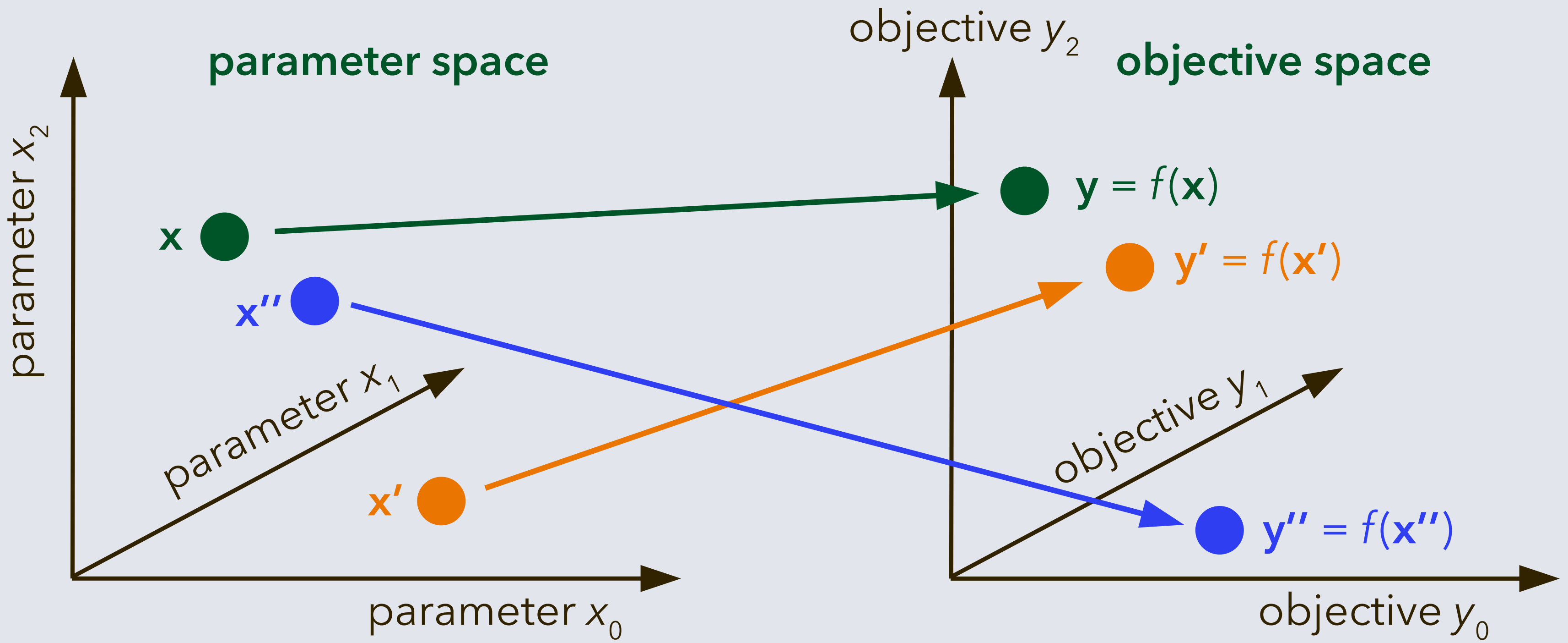
Formalization of optimization problems



Design choices are made by selecting a point in parameter space, but they are **evaluated in objective space**, by comparing the outcomes for $y_0, \ldots, y_{n-1}$.

The optimization problem is defined by a function f that maps a list of parameters x = $[x_0, \ldots, x_{m-1}]$ to the outcome for the objectives $\mathbf{y} = f(\mathbf{x}) = [y_0, \ldots, y_{n-1}]$.

Formalization of optimization problems

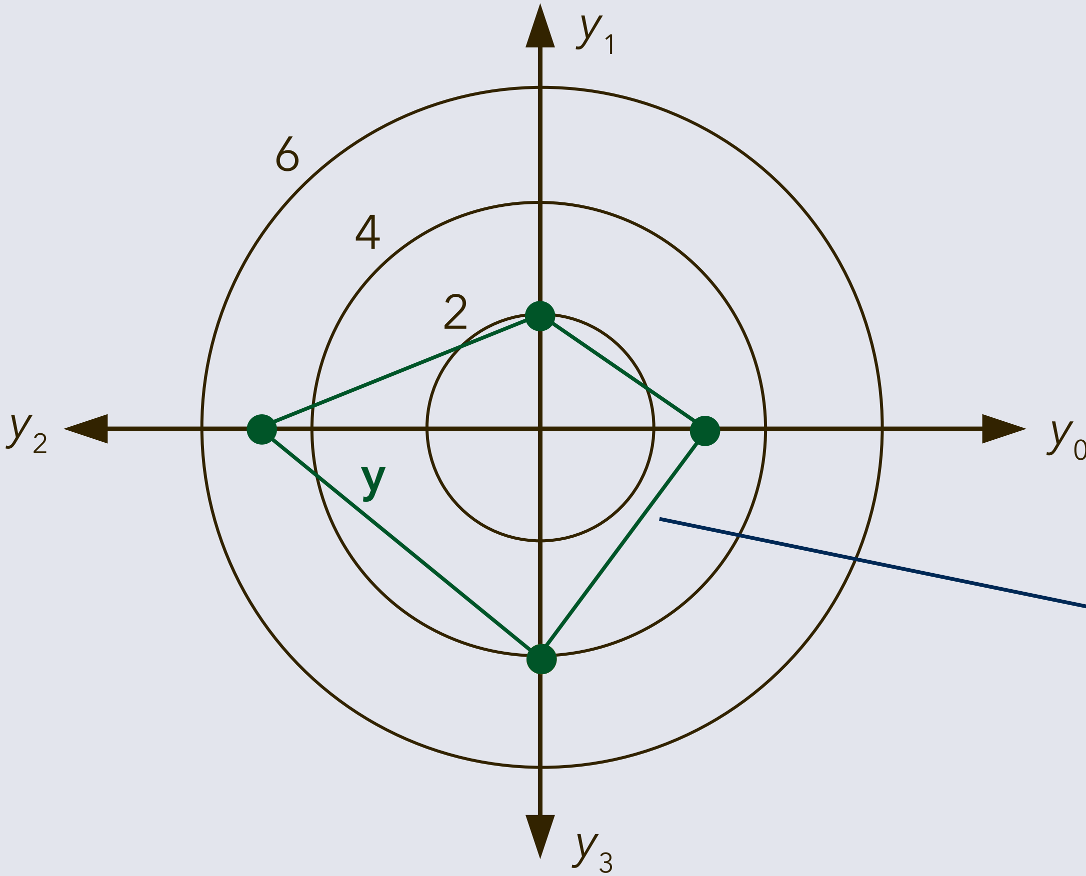


Design choices are made by selecting a point in parameter space, but they are evaluated in objective space, by comparing the outcomes for $y_0, \ldots, y_{n-1}$.

How are the design choices evaluated? How can we compare $\mathbf{y} = f(\mathbf{x})$, $\mathbf{y}' = f(\mathbf{x}')$, and $\mathbf{y}'' = f(\mathbf{x}'')$, and decide **which is better**, when there are **conflicting criteria**?

Assessment of solutions using multiple criteria

Spider diagrams are often used to visualize points in objective space.



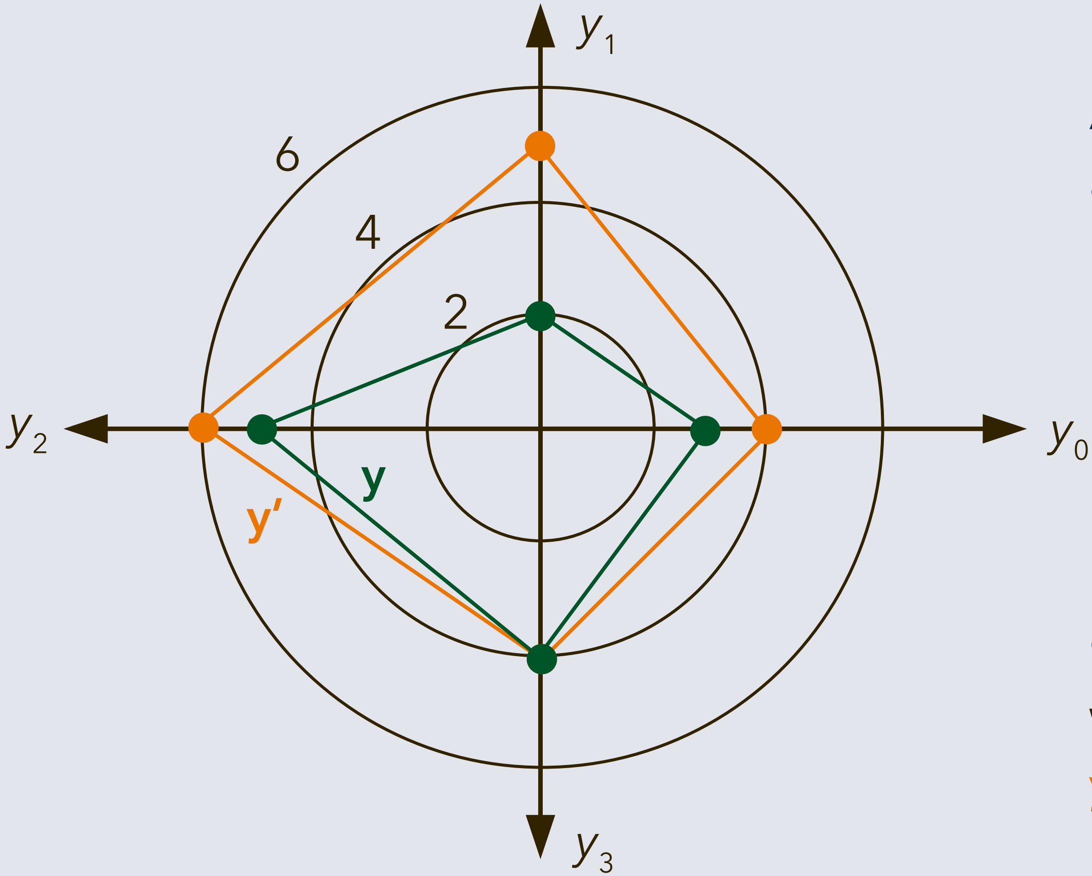
Assume that y_0, y_1, y_2, and y_3 are all minimization objectives.

$\mathbf{y} = [3, 2, 5, 4]$

Each polygon in a spider diagram represents **one point** in objective space, in this case **y**.

Assessment of solutions using multiple criteria

Spider diagrams are often used to visualize points in objective space.



Assume that y_0, y_1, y_2, and y_3 are all minimization objectives.

y = [3, 2, 5, 4]

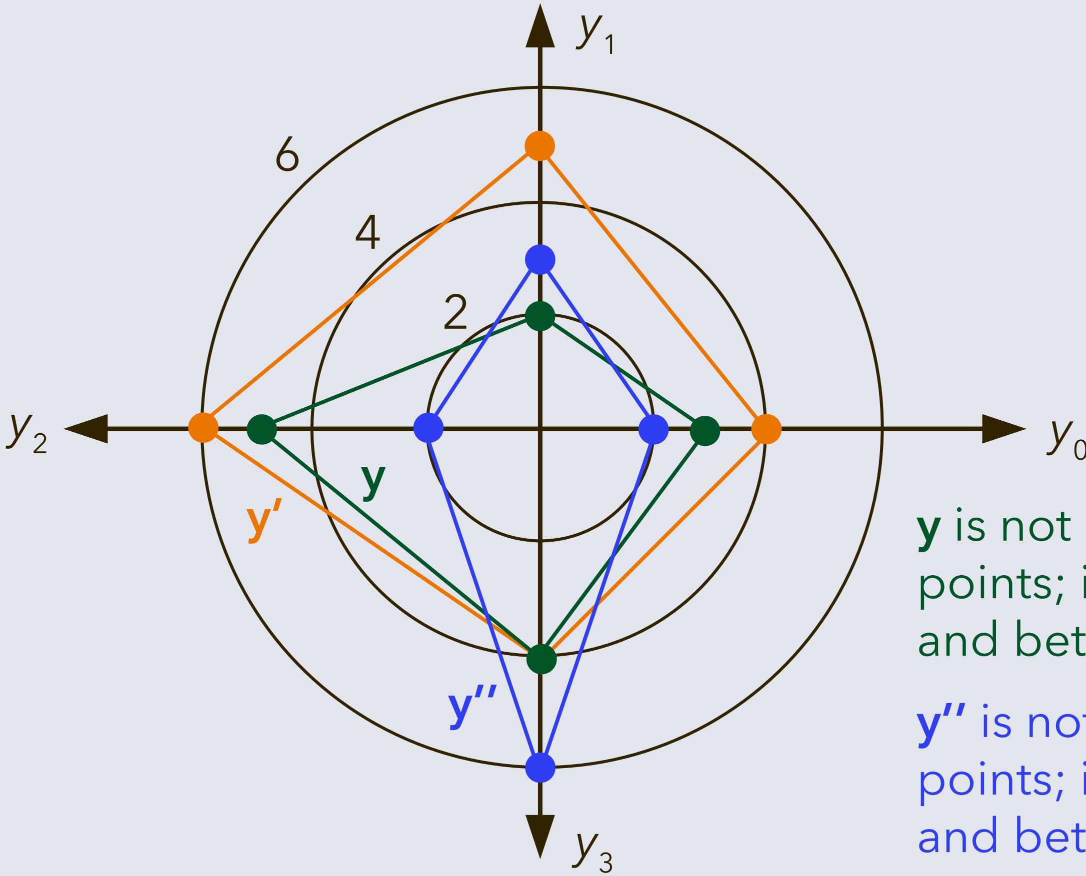
y' = [4, 5, 6, 4]

Note: **y** and **y'** perform equally in criterion y_3, and in the three other criteria, **y** outperforms **y'**.

We say: y dominates y'. Since **y'** is dominated, it cannot be a rational choice to select **y'**.

Assessment of solutions using multiple criteria

Spider diagrams are often used to visualize points in objective space.



Assume that y_0, y_1, y_2, and y_3 are all minimization objectives.

y = [3, 2, 5, 4]

y′ = [4, 5, 6, 4]

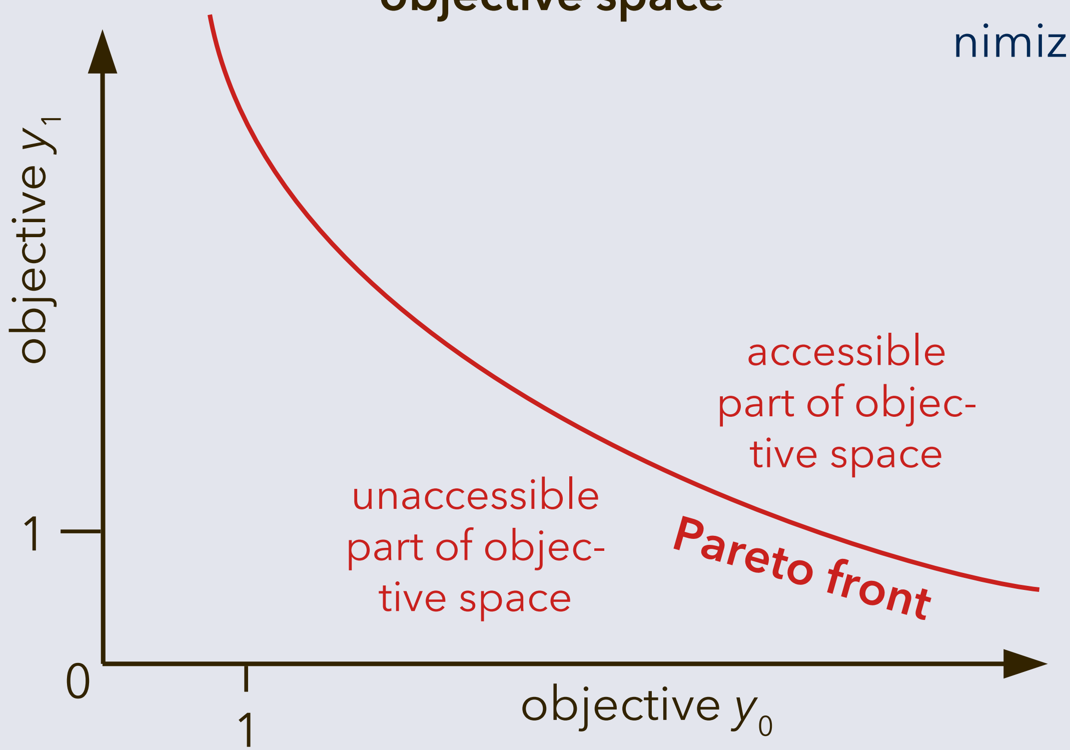
y′′ = [2, 3, 2, 6]

y is not dominated by any of the other points; it is better than **y′** in three criteria, and better than **y′′** in two criteria.

y′′ is not dominated by any of the other points; it is better than **y′** in three criteria, and better than **y** in two criteria.

The Pareto front

An accessible point in objective space belongs to the **Pareto front** whenever it is **not dominated** by any other accessible point in objective space.



Assume that there are two minimization objectives, y_0 and y_1.

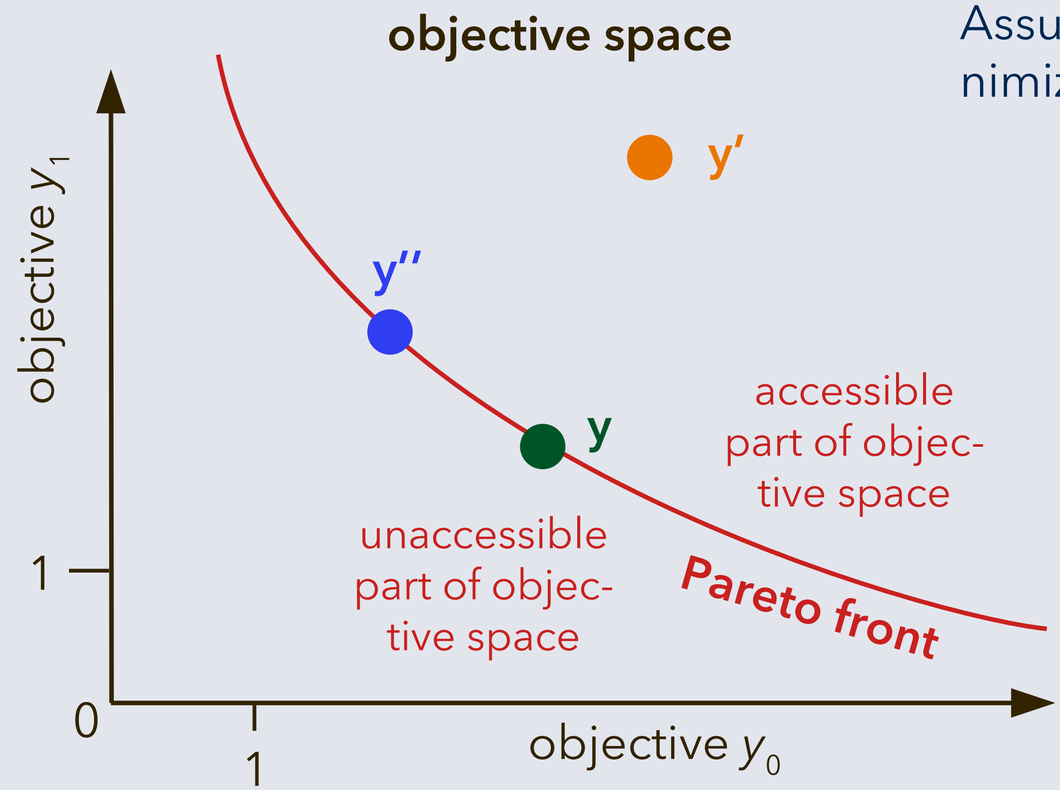
$\mathbf{y} = [3, 2]$

$\mathbf{y'} = [4, 5]$

$\mathbf{y''} = [2, 3]$

The Pareto front

An accessible point in objective space belongs to the **Pareto front** whenever it is **not dominated** by any other accessible point in objective space.



Assume that there are two minimization objectives, y_0 and y_1.

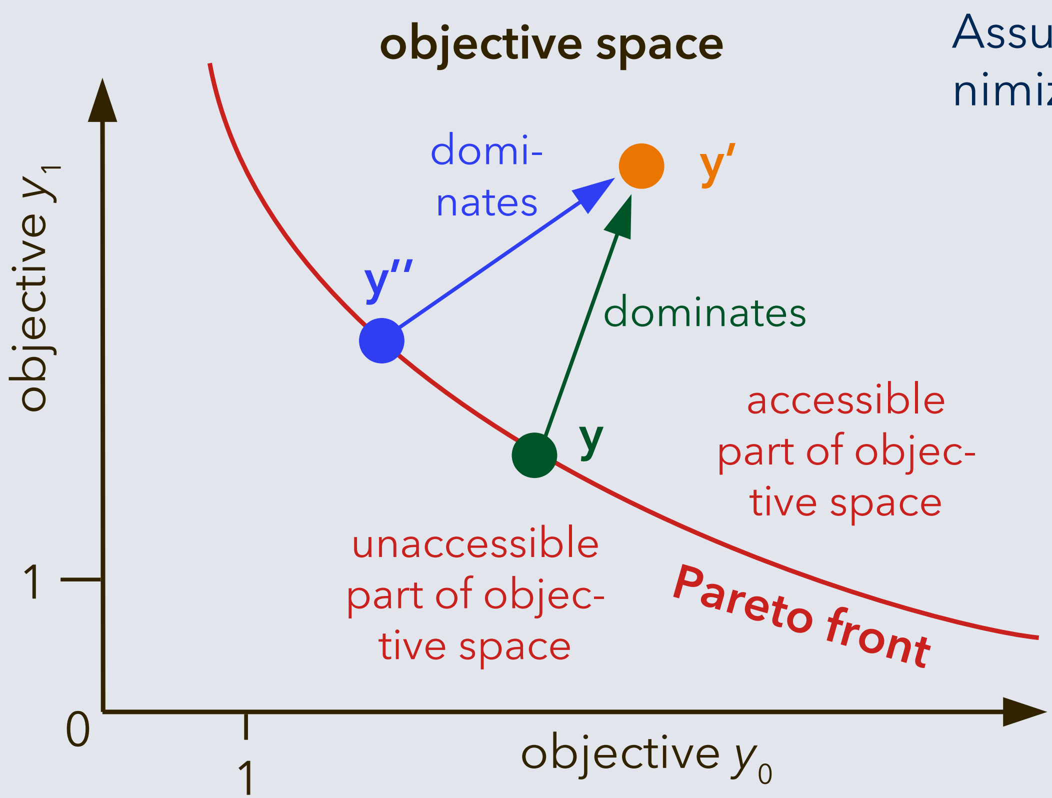
y = [3, 2]

y′ = [4, 5]

y″ = [2, 3]

The Pareto front

An accessible point in objective space belongs to the **Pareto front** whenever it is **not dominated** by any other accessible point in objective space.



Assume that there are two minimization objectives, y_0 and y_1.

y = [3, 2]
y′ = [4, 5]
y″ = [2, 3]

y and **y″** are rational compromises between the two objectives.

y′ is not, because it is dominated by other accessible points.

Pareto optimality

An accessible point in objective space belongs to the **Pareto front** whenever it is **not dominated** by any other accessible point in objective space.

A point $\mathbf{y}$ in objective space **dominates** another point $\mathbf{y}'$ if there is at least one criterion by which $\mathbf{y}$ is better than $\mathbf{y}'$, but no criterion by which $\mathbf{y}'$ is worse. Consequently, if $\mathbf{y}$ is accessible, it can never be rational to select $\mathbf{y}'$.

Not all points in objective space are accessible. A point $\mathbf{y}$ in objective space is **accessible** whenever there is a point $\mathbf{x}$ in parameter space such that $f(\mathbf{x}) = \mathbf{y}$.

Pareto optimality

An accessible point in objective space belongs to the **Pareto front** whenever it is **not dominated** by any other accessible point in objective space.

A point $\mathbf{y}$ in objective space **dominates** another point $\mathbf{y}'$ if there is at least one criterion by which $\mathbf{y}$ is better than $\mathbf{y}'$, but no criterion by which $\mathbf{y}'$ is worse. Consequently, if $\mathbf{y}$ is accessible, it can never be rational to select $\mathbf{y}'$.

Not all points in objective space are accessible. A point $\mathbf{y}$ in objective space is **accessible** whenever there is a point $\mathbf{x}$ in parameter space such that $f(\mathbf{x}) = \mathbf{y}$.

Pareto optimal solutions are potential rational compromises between criteria.

Pareto optimality

An accessible point in objective space belongs to the **Pareto front** whenever it is **not dominated** by any other accessible point in objective space.

A point **y** in objective space **dominates** another point **y'** if there is at least one criterion by which **y** is better than **y'**, but no criterion by which **y'** is worse. Consequently, if **y** is accessible, it can never be rational to select **y'**.

Not all points in objective space are accessible. A point **y** in objective space is **accessible** whenever there is a point **x** in parameter space such that $f(\mathbf{x}) = \mathbf{y}$.

To remember:

- The Pareto front is situated in objective space (not in parameter space).
- It contains all the accessible points that are not dominated.
- A solution **x** (*i.e.*, a parameter choice, or a point in parameter space) is Pareto optimal if and only if $\mathbf{y} = f(\mathbf{x})$ is on the Pareto front.

Pareto optimality

An accessible point in objective space belongs to the **Pareto front** whenever it is **not dominated** by any other accessible point in objective space.

A point y ... st one criterion ... e. Consequ...

Not all p... pace is **accessib**...) = **y**.

To reme...

- T... space).
- It...
- A solution **x** (*i.e.*, a parameter choice, or a point in parameter space) is Pareto optimal if and only if **y** = f(**x**) is on the Pareto front.

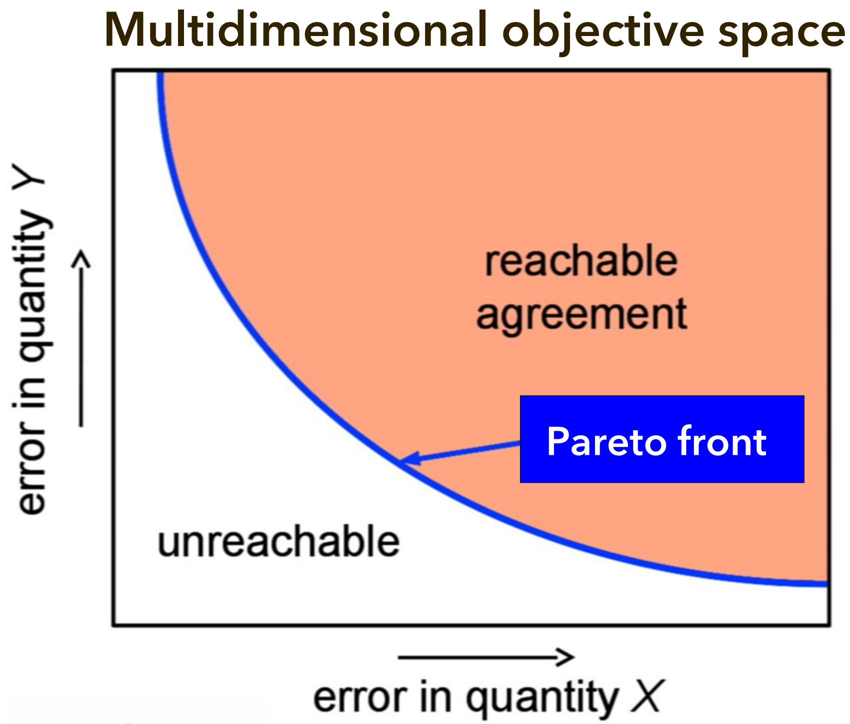
Terminology

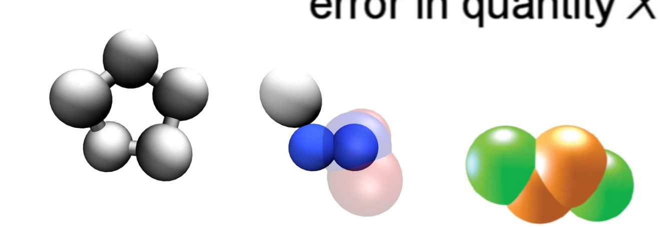
A solution or design choice **x**, *i.e.*, a point in parameter space, is sometimes also called a **parameterization**. Strictly speaking, parameterization is the process of selecting **x**.

It is bad practice to say "the solution **x** is on the Pareto front," since the **Pareto front** is situated in objective space. Instead, we say "the point f(**x**) is on the Pareto front."

However, both **x** and f(**x**) may be called "**Pareto optimal**."

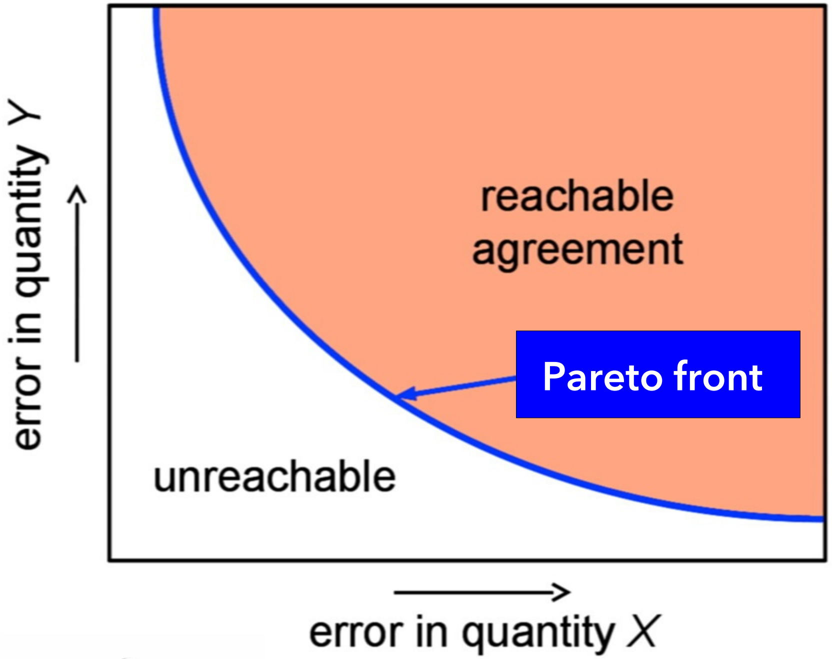
Example from research practice

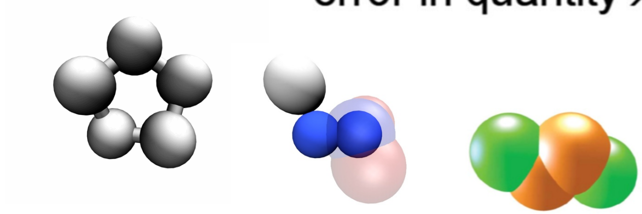




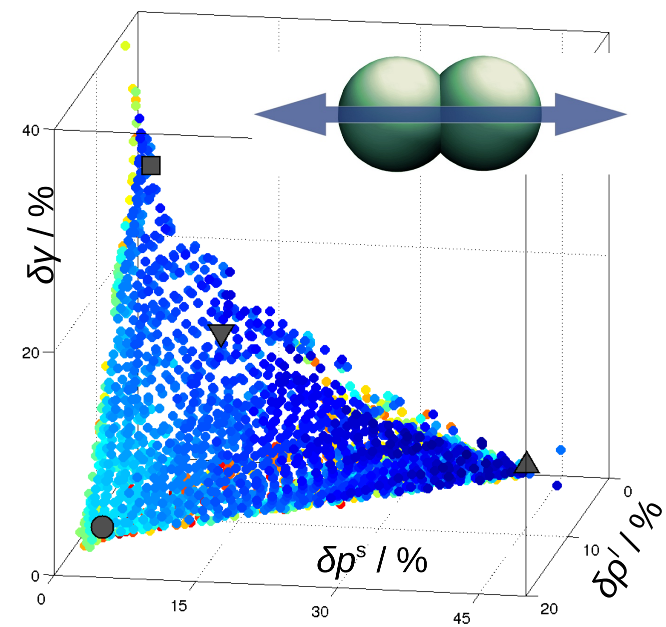
Example from research practice

Multidimensional objective space



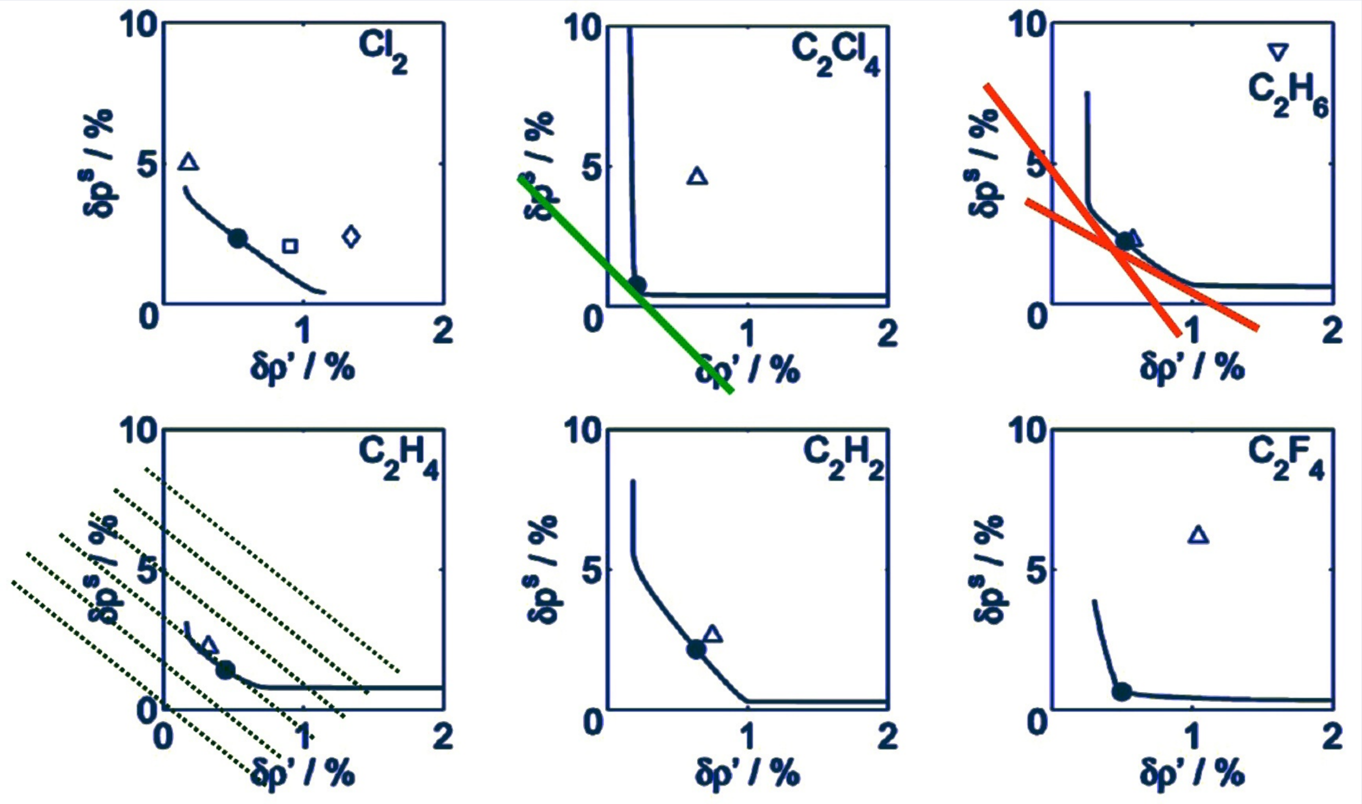


2CLJQ molecular models of carbon dioxide



Resilient compromises between objectives

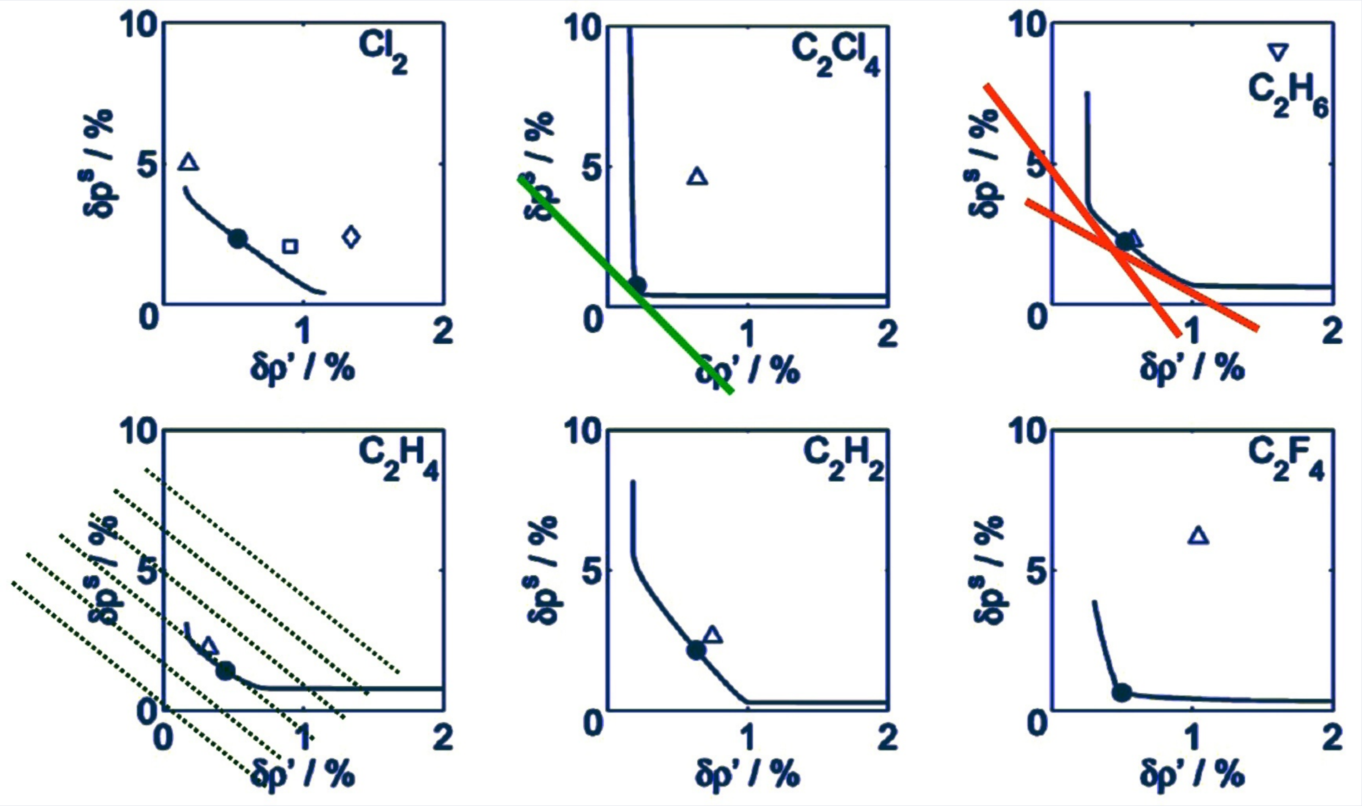
(Example: Two-criteria optimization of molecular models.)



A **Pareto knee** is a highly curved region on the Pareto front.

Resilient compromises between objectives

The viability of models close to a **Pareto knee** is comparably resilient even when priorities shift. Example: Two-criteria optimization of molecular models.



A **Pareto knee** is a highly curved region on the Pareto front.

In general, a systematic **exploration of the Pareto front** is needed to find such regions.

University of Central Lancashire
UCLan

Terminology and building a glossary

CO3519

26th October 2021

University of Central Lancashire
UCLan

CO3519
Artificial Intelligence

Parameters and objectives
Multicriteria optimization
Terminology and building a glossary

Where opportunity creates success