Multi-objective (criteria) optimization problem (MOOP)

Single objective optimization

$$\min F(x) = \begin{Bmatrix} f_1(\bar{x}) \\ f_2(\bar{x}) \\ \vdots \\ f_k(\bar{x}) \end{Bmatrix}$$

k objectives

s.t. $\bar{x} \in S$

$\bar{x} = (x_1, .. x_n)$

n optimization variables

Solution space

$g(\bar{x}) \leq 0$

$h(\bar{x}) = 0$

$x_i^{low} \leq x_i \leq x_i^{up}$

$$\min f(x)$$

s.t. $\bar{x} \in S$

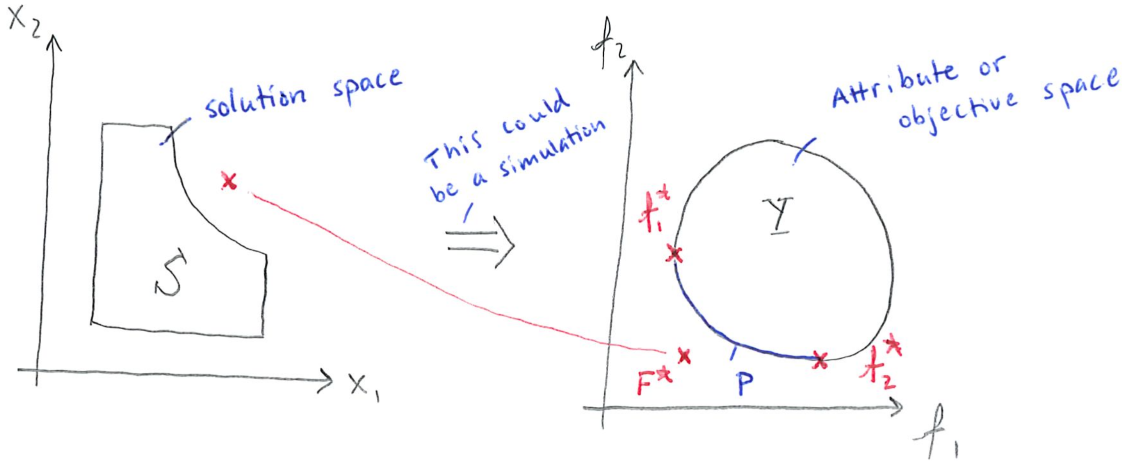
$\bar{x} = (x_1, .. x_n)^t$

In a strict mathematical sense, the MO-formulation is not valid as min $F(x)$ Lacks clear meaning as it is not obvious how different designs should be ordered, i.e. is $F(\bar{x}_1)$ better than $F(\bar{x}_2)$?

MOO is about ordering the set $F(\bar{x})$ so that min $F(\bar{x})$ will have a clear meaning

This ordering may include the subjective decision maker (DM)

Graphical Representation for a problem with two optimization variables and two objectives.



f_1^* and f_2^* are the individual optima for the two objective functions

F^* is a point in attribute space where $f_1 = f_1^*$ and $f_2 = f_2^*$
As it rarety is in the feasible region it is called the utopian solution

P is the Pareto Front. In the 2D case it is a curve between f_1^* and f_2^*

* The Pareto front expresses the tradeoff between the optimal solutions

* Different points on the Pareto Front might be the best point depending on the DM's preferenses

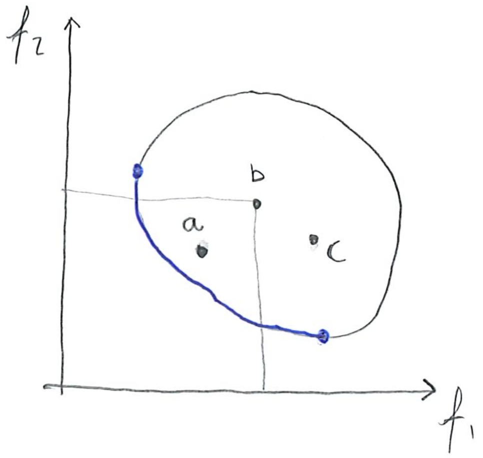
* A point on the front is a rational choice for the final design. A point inside is not

* In MOO, one goal is to identify as much of the front as possible.

* In SOO, the objectives are aggregated (put together) into an overall objective function. The result is then one point on the Pareto front

A central part of MOO is Pareto dominance

Graphical Explanation:



a dominates b
a dominates c

b & c are dominated by a
b & c do not dominate each other

Mathematical Explanation
$a \succ b$ if:

$\forall i \in \{1, 2, \ldots, k\}: f_i(a) \leq f_i(b)$ and $\exists j \in \{1, 2, \ldots, k\}: f_j(a) < f_j(b)$

Plane text explanation
a dominates b if a is as good as b for all k objectives and there exists at least one objective where a is better

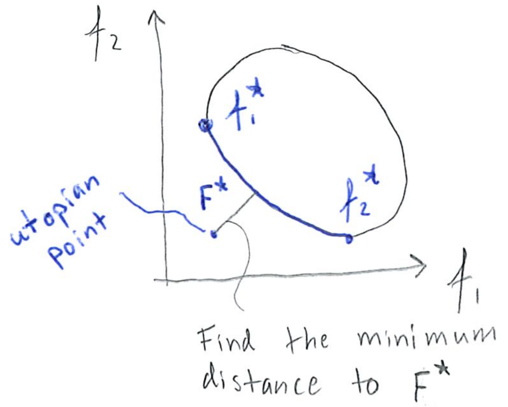
Common sense:
If a dominates b it is rational to choose a over b
If none dominates the other, a subjective judgement is needed

Articulation of preference info

There are four different occasions when the D.M. can articulate his/her preferences among the objectives:

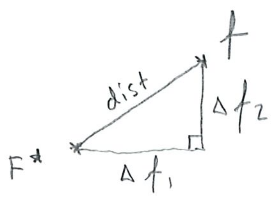
* Never (the DM can't affect the outcome of the optimization)
* Before the optimization (the DM aggregates the objectives to 1 fcn)
* During the optimization (the DM changes the obj. fun. during the optimization)
* After the optimization (first we optimize - then the DM selects one optimal solution)

Min-max formulation



$$\min \left[\sum_{j=1}^{k} \left(\frac{f_j(\bar{x}) - f_j^*}{f_j^*} \right)^p \right]^{\frac{1}{p}}$$

with $p = 2$ it is the Euclidean distance

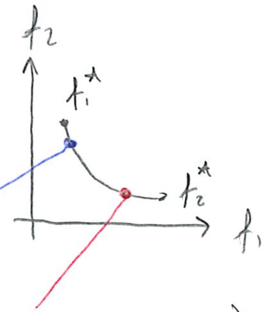


$$\text{dist} = \sqrt{(\Delta f_1)^2 + (\Delta f_2)^2}$$

Weighted Sum

$$\min \sum_{j=1}^{k} \lambda_j f_j(\bar{x})$$

s.t. $\bar{x} \in S$



Usually $\lambda \in R^k \mid \lambda_j > 0 \ , \ \sum \lambda_j = 1$

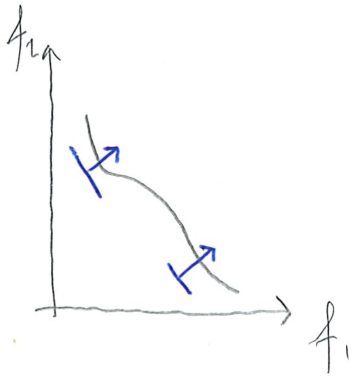
(e.g. $\lambda_1 = 0.9$ & $\lambda_2 = 0.1$ or $\lambda_1 = 0.2$ & $\lambda_2 = 0.8$)

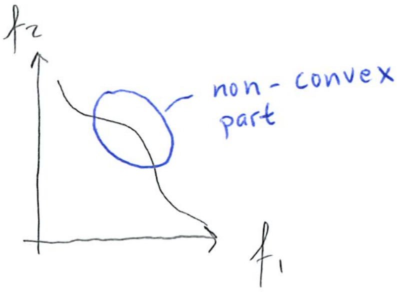
$F(\bar{x}) = 0.9 f_1(\bar{x}) + 0.1 f_2(\bar{x})$ $\qquad$ $F(\bar{x}) = 0.2 f_1(\bar{x}) + 0.8 f_2(\bar{x})$

Then you usually often have to normalize the objectives

$$\min \sum_{j=1}^{k} \lambda_j \frac{f_j(\bar{x})}{f_j^*} \quad \text{or} \quad \min \sum_{j=1}^{k} \lambda_j \frac{f_j(\bar{x})}{f_{j0}}$$

Linear combinations are highly unlikely to find points on the non-convex part of the Pareto Front





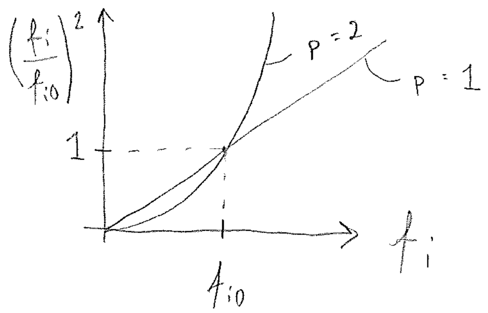
Non-Linear Combinations

One formulation of many possible

$$\min \sum_{j=1}^{k} \left(\frac{f_j(\bar{x})}{f_{j0}}\right)^p$$

$$\text{s.t.} \quad \bar{x} \in S$$

Ex: $\min \; F(x) = \left(\dfrac{f_1}{f_{10}}\right)^2 + \left(\dfrac{f_2}{f_{20}}\right)^2$



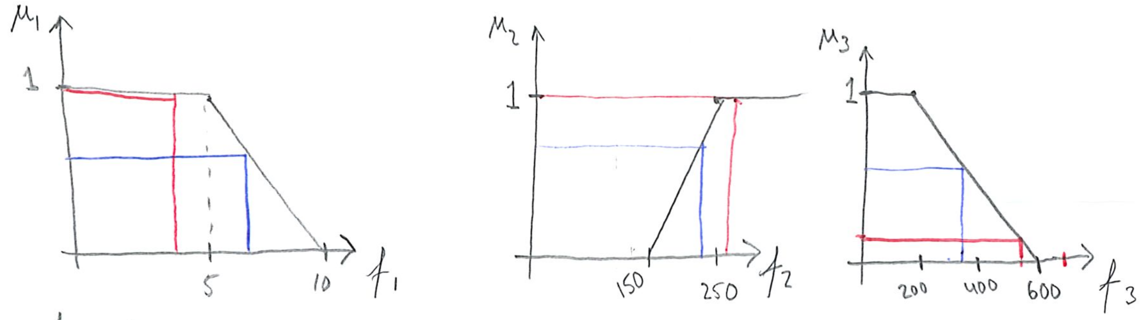
- The objectives are normalized
- p expresses how much a bad objective should penalize a solution
- The curvature should express the decision maker's preferences

Fuzzy Logic Approach

* Fuzzy Logic is a multi-valued logic where a statement could be simultaneously partly true and partly false
* The membership function μ expresses the thruthfulness of a statement

$$F_{Fuzzy}(\bar{x}) = \prod_{j=1}^{k} \mu_j(f_j(\bar{x}))$$

Sports Car Example



f_1 = 0 – 100 acceleration time [s]

f_2 : top speed [km/h]

f_3 = cost

Car 1 : acc : 4s, top speed = 260, cost = 550 $\Rightarrow$ $\mu_1 = 1$, $\mu_2 = 1$, $\mu_3 = 0.2$ $\Rightarrow$ $F_{fuzzy}(\bar{x}) = 1 \cdot 1 \cdot 0.2 = 0.2$

Car 2 : acc = 7s, top speed = 230, cost = 350 $\Rightarrow$ $\mu_1 = 0.6$, $\mu_2 = 0.7$, $\mu_3 = 0.7$ $\Rightarrow$ $F_{fuzzy}(\bar{x}) = 0.6 \cdot 0.7 \cdot 0.7 \approx 0.3$

$\Rightarrow$ Car 2 > Car 1

Goal Programming

- Assign a goal to each objective

  for example $f_1(\bar{x}) \geq 10$

Even though goal programming might sound attractive it has some problems:

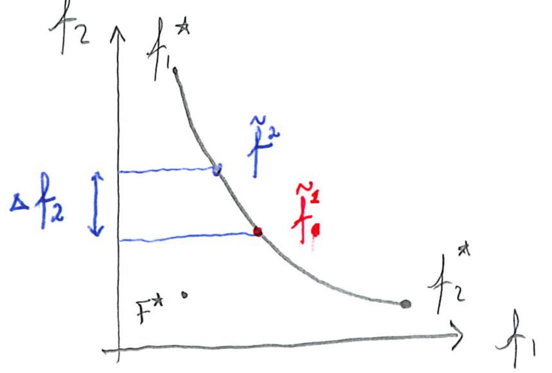
- The subobjectives are not really optimized
- Setting the goal level for one objective might affect other objectives in unforseen ways
- When you are trying to get as close as possible to the goal area you are not weighting the objectives properly

STEM-Method

A weighted sum method where the solution space is gradually reduced.

- Formulate the problem as min $\lambda_1 f_1 + \lambda_2 f_2$ with initial guesses on λ_1 and λ_2.
- Solve the problem to obtain $\tilde{f}$
- Compare $\tilde{f}$ to F^*
- If not acceptable determine a relaxation in one objective
- Insert $f_j \leq \tilde{f}_j + \Delta f_j$ as a new constraint and set $\lambda_j = 0$
- Solve the new problem
- Satisfied?

Graphical Explanation



- Solve min $\lambda_1 f_1 + \lambda_2 f_2 \Rightarrow \tilde{f}^1$
- $\tilde{f}^1$ is compared to F^* and the DM states that $\tilde{f}_1^1$ is to large
- The DM is prepared to pay Δf_2 to obtain a better f_1 value

- Reformulate the problem as

$$\begin{aligned} \min \quad & f_1 \\ \text{s.t.} \quad & x \in S \\ & f_2 \leq \tilde{f}_2 + \Delta f_2 \end{aligned}$$

- Solve the problem and obtain $\tilde{f}^2$
- Satisfied?

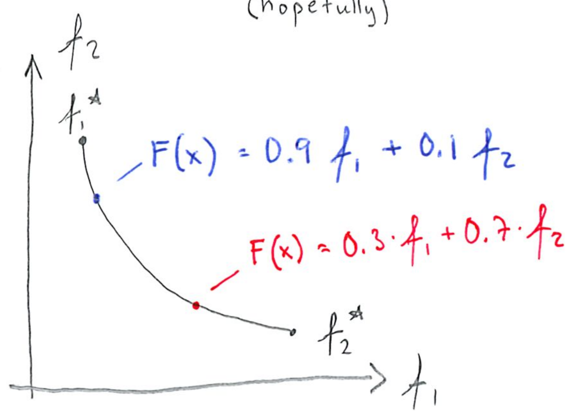
Weighted sum method

$$\min \quad F(x) = \lambda_1 f_1(\bar{x}) + \lambda_2 f_2(\bar{x}) + \ldots + \lambda_k f_k(\bar{x})$$

$$\text{s.t.} \quad \bar{x} \in S$$

$$\lambda_i \geq 0, \quad \sum_{i=1}^{k} \lambda_i = 1$$

Perform optimizations with different λ_i. Each optimization yields a new point on the Pareto front (hopefully)



You often need to normalize the objectives

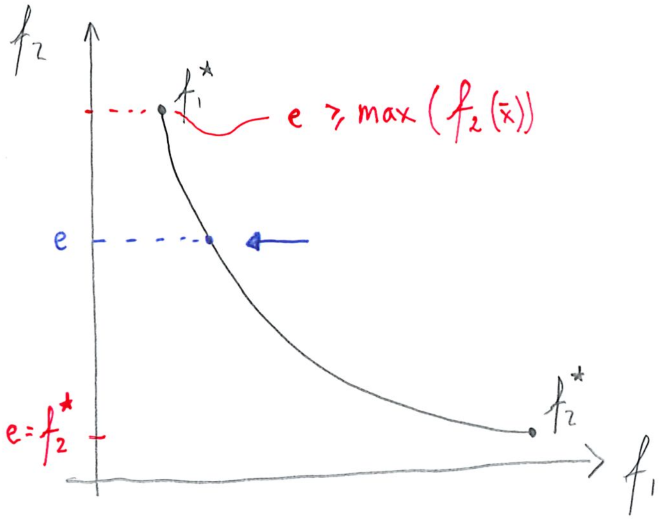
ex: weight ≈ 10 vs stress $\approx 10^6$

E - Constraint Method

$$\min \ f_1(\bar{x})$$

$$\text{s.t.} \quad \bar{x} \in S$$

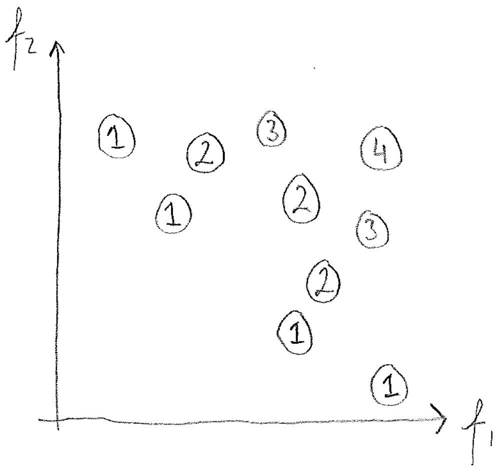
$$f_2 \le e, \quad f_2^* < e < \infty$$



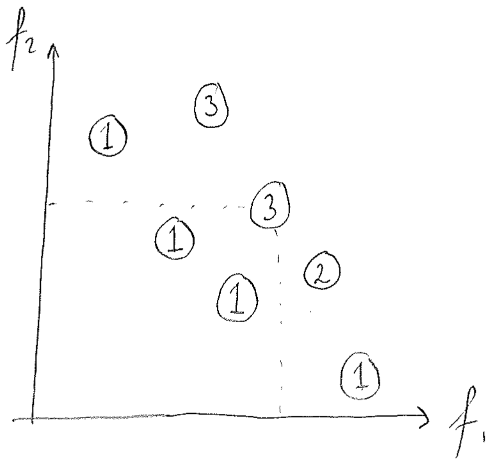
Different values of e results in different points on the Pareto front

MOGA

Algorithms that try to find as much of the Pareto front as possible in one run



Non-dominated Sorting



Dominated sorting

no = no_of_better_points + 1

Algorithm outline

- Init population
- Rank pop according to Pareto dominance
- Select parents
- Create children
- Calculate $f_1, f_2, \dots, f_k$
- Insert children
- Rank population
- Share fitness (clustering etc.)
- Stop?

Spread the individuals evenly on the Pareto front