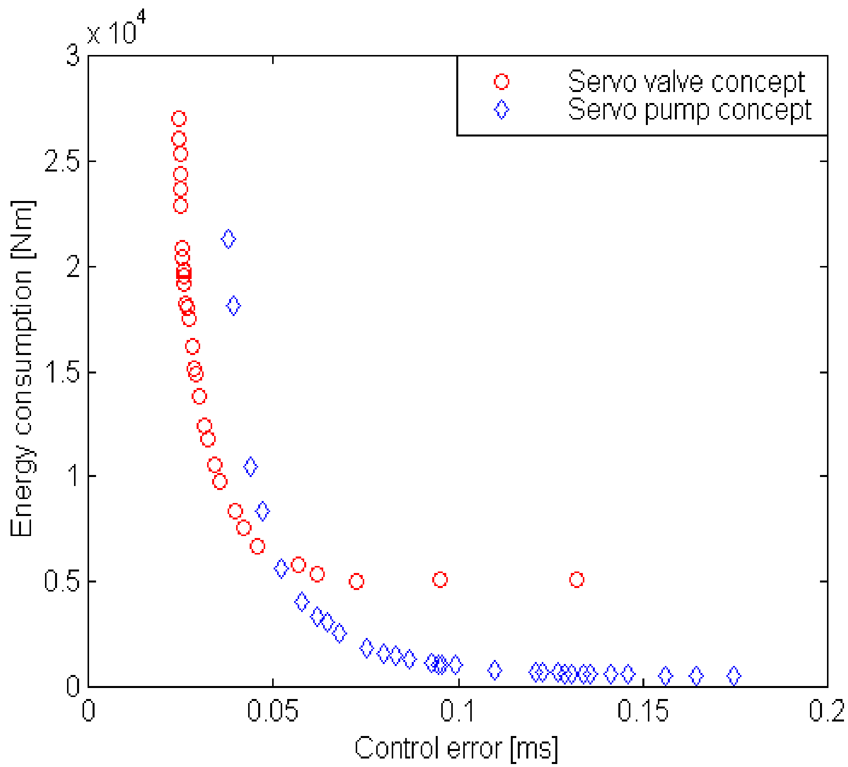
Multi-objective optimization

All design problems are in reality multi-objective to their nature.



Contents

- Mathematical formulation
- How to aggregate multiple objectives?
  - No articulation of preferences
  - Aggregation of many objectives to one objective function
  - Iterative methods
  - Pareto optimization
  - Multi-objective genetic algorithms

Multi-objective formulation

$$\min \mathbf{F}(\mathbf{x}) = \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x}) \right)^T$$

$$s.t. \;\; \mathbf{x} \in S$$

$$\mathbf{x} = \left(x_1, x_2, \ldots, x_n \right)^T$$

k = number of objectives

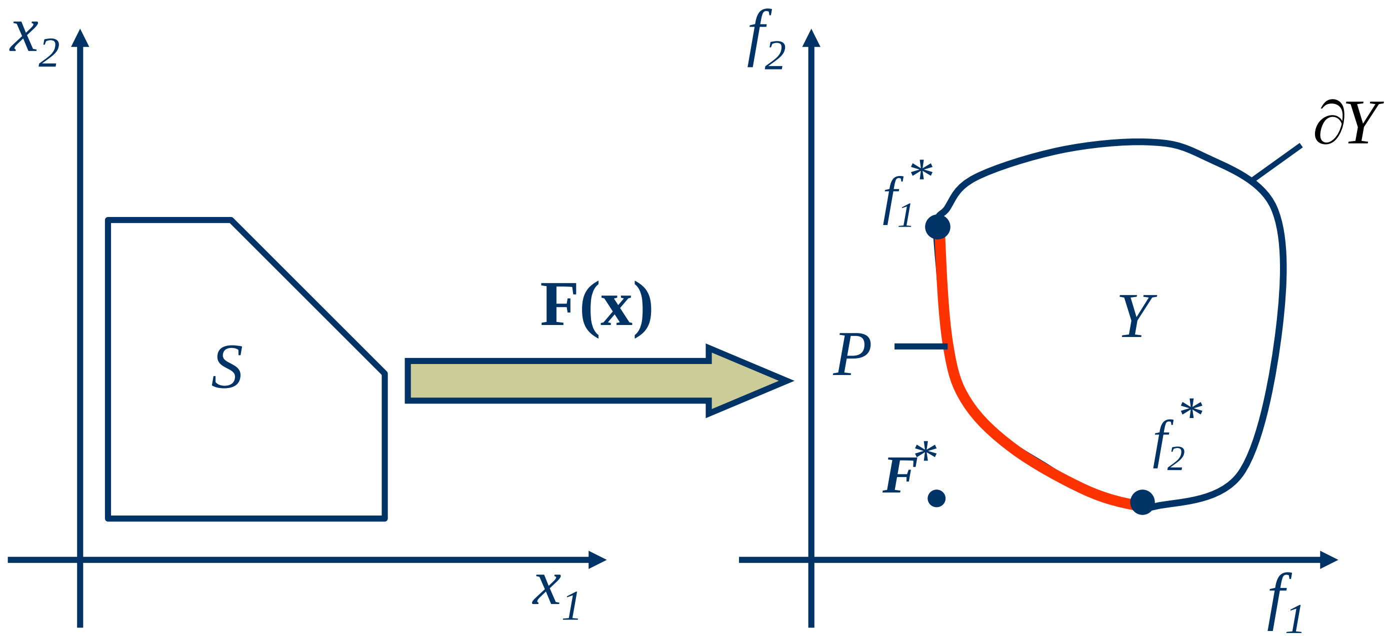
n = number of parameters

f_i = system characteristic, or sub-objective

x_i = optimization variables

S = solution space

Problem visualization



S = parameter space

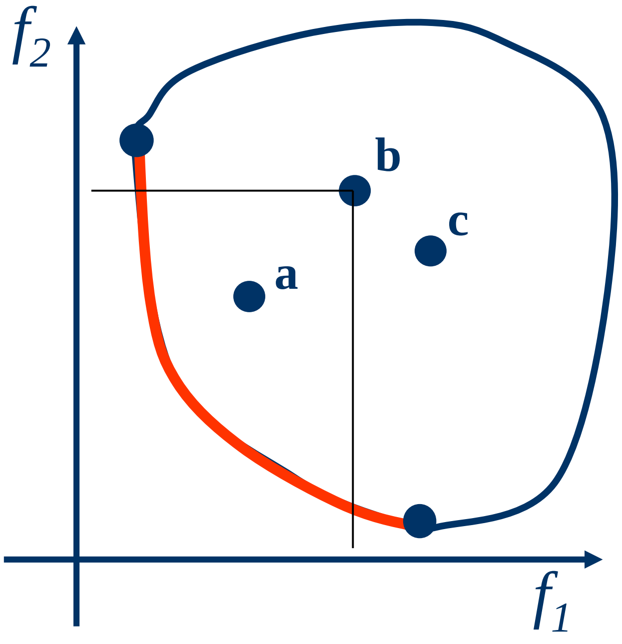
Y = objective or attribute space

f_i^*= individual optima

F*= utopian solution

P = Pareto optimal front

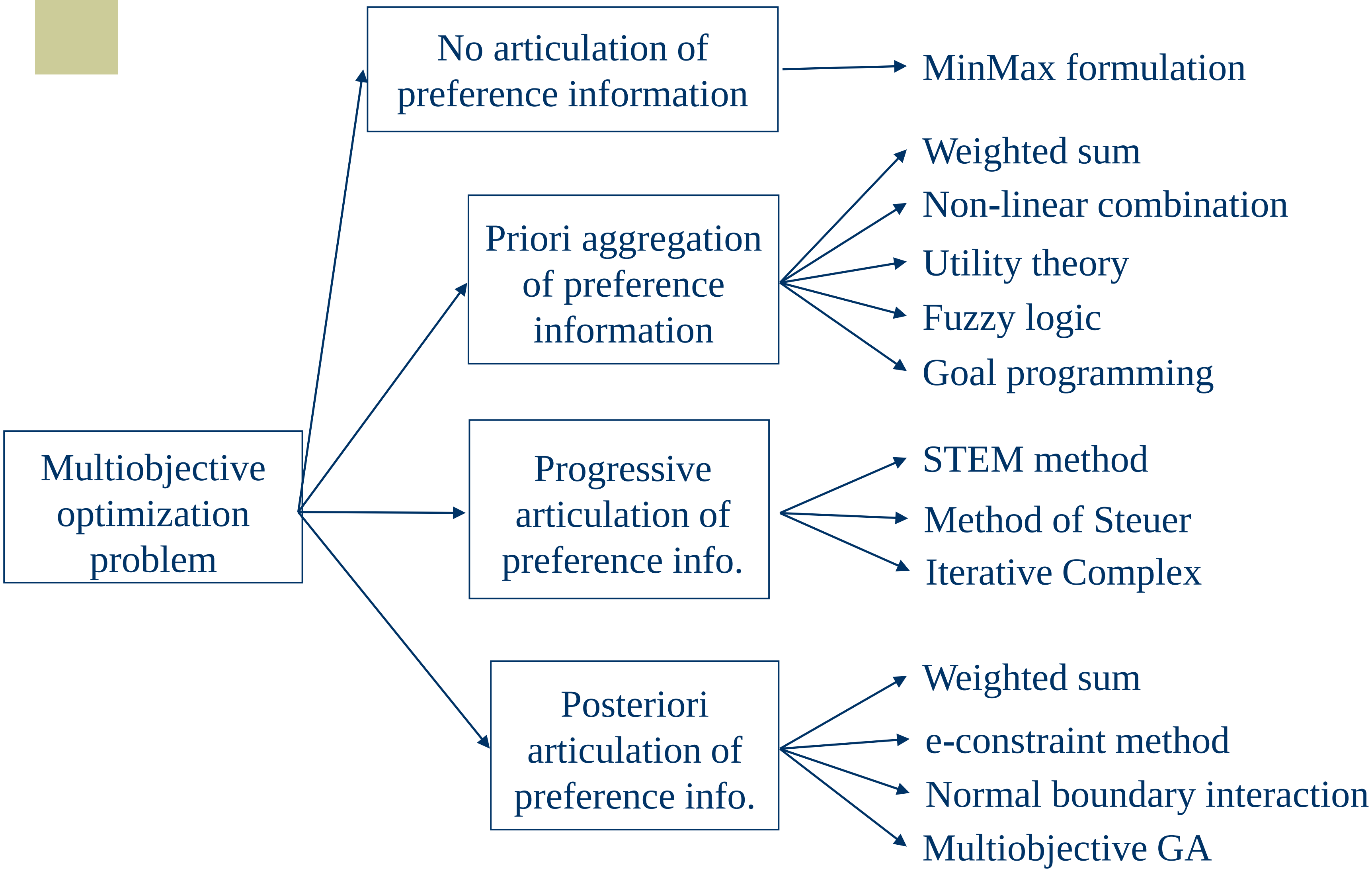
Pareto dominance



a is said to dominate b, ($\mathbf{a} \succ \mathbf{b}$), if:

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

Articulation of preference info.



Min-max formulation

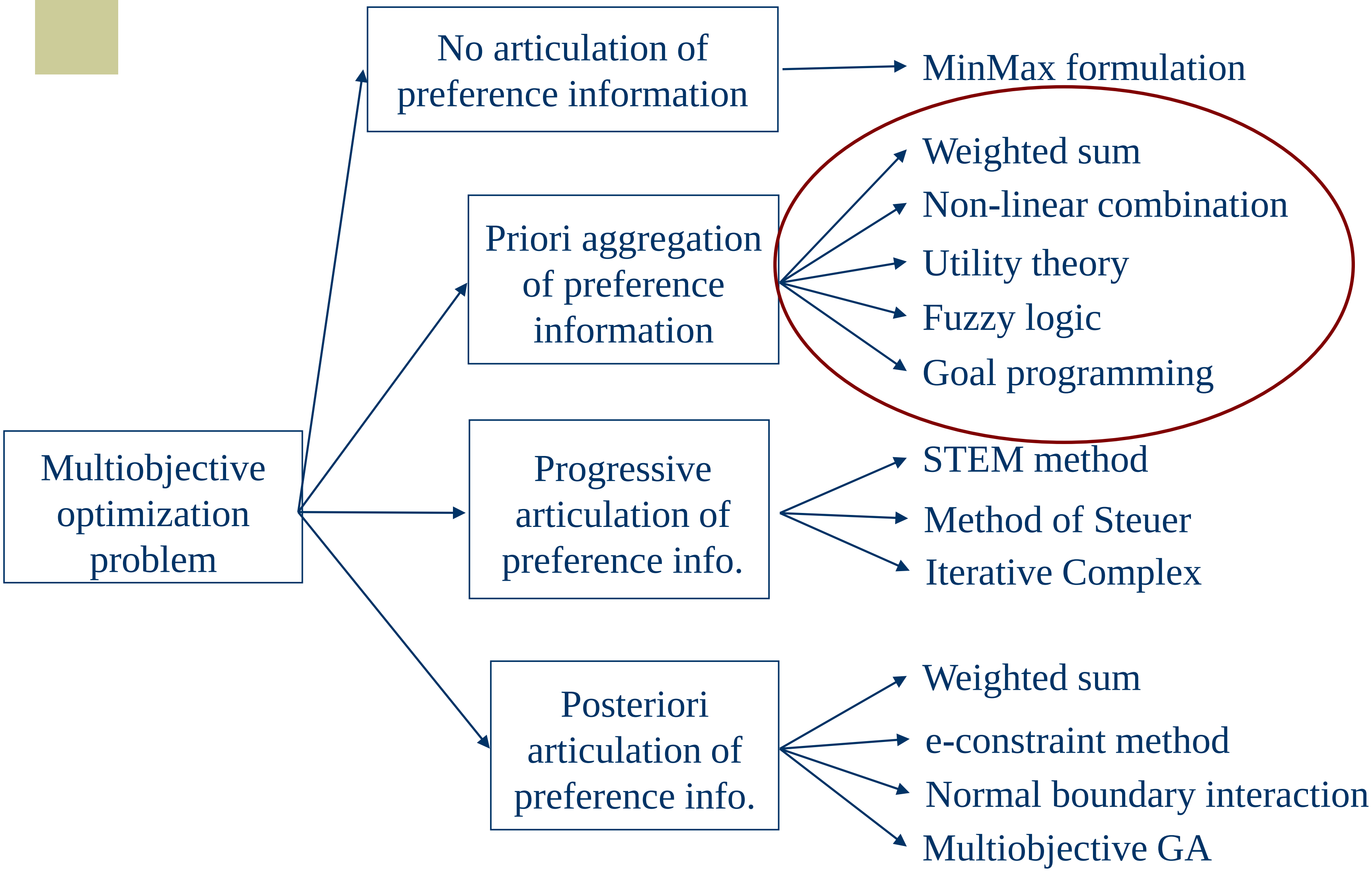
Minimize the relative distance from a candidate solution to the utopian solution $\mathbf{F}^*$.

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

$$s.t. \quad \mathbf{x} \in S$$

$$1 \le p \le \infty$$

Articulation of preference info.



Weighted sum

The objective is formulated as a weighted sum of all objectives.

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

$$s.t. \quad \mathbf{x} \in S$$

$$\lambda \in R^k \,\middle|\, \lambda_i > 0, \sum \lambda_i = 1$$

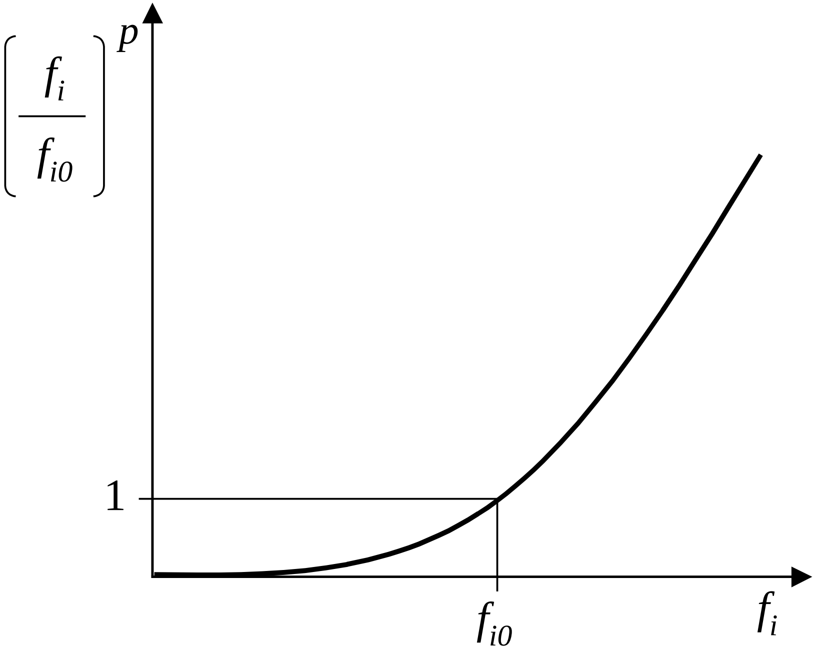
Weighted sum: *comments*

- It is the most easy way to aggregate the objectives
- The procedure of determining the weights is ad-hoc
- The relation between the weights and the optimum are unknown beforehand.
- Linear combinations of the objectives can not find points on non- convex parts of the Pareto front

Non-linear combinations

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

$$s.t. \quad \mathbf{x} \in S$$



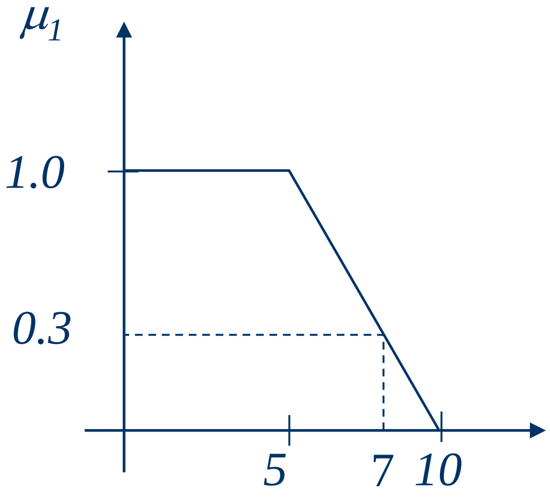
Non-linear combinations: *comments*

- The objectives are normalized
- The coefficient p determines how much a bad value should penalise the solution
- The curvature should represent the decision-makers preferences.

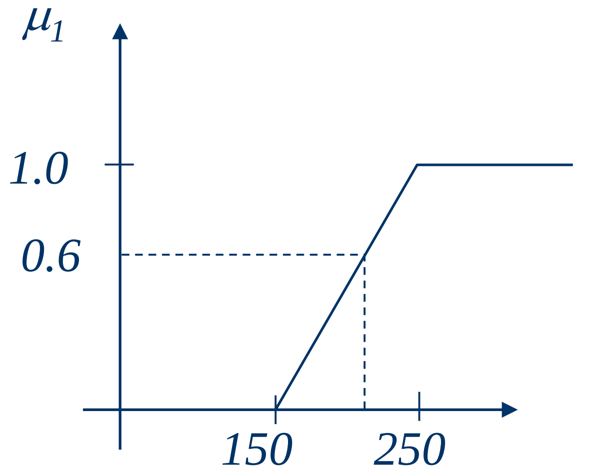
Fuzzy logic approach

- Fuzzy logic is a multi-valued logic.
- Statements could simultaneously be partly true and partly false.
- The member function μ expresses the truthfulness of a statement.
- The value of a sub-objective *f* is fuzzyfied by μ.

Fuzzy logic – *member functions*



Acc time 0-100 km/h



Top speed [km/h]

A sports car example

Fuzzy logic: *aggregating objectives*

$$F_{fuzzy}(\mathbf{x}) = \prod_{i=1}^{k} \mu_i\left(f_i(\mathbf{x})\right)$$

$$\max \; F_{fuzzy}(\mathbf{x})$$

$$s.t. \;\; \mathbf{x} \in S$$

Goal programming

In goal programming (GP) the objectives are formulated as goal criteria that the decision maker wants each objective to possess.

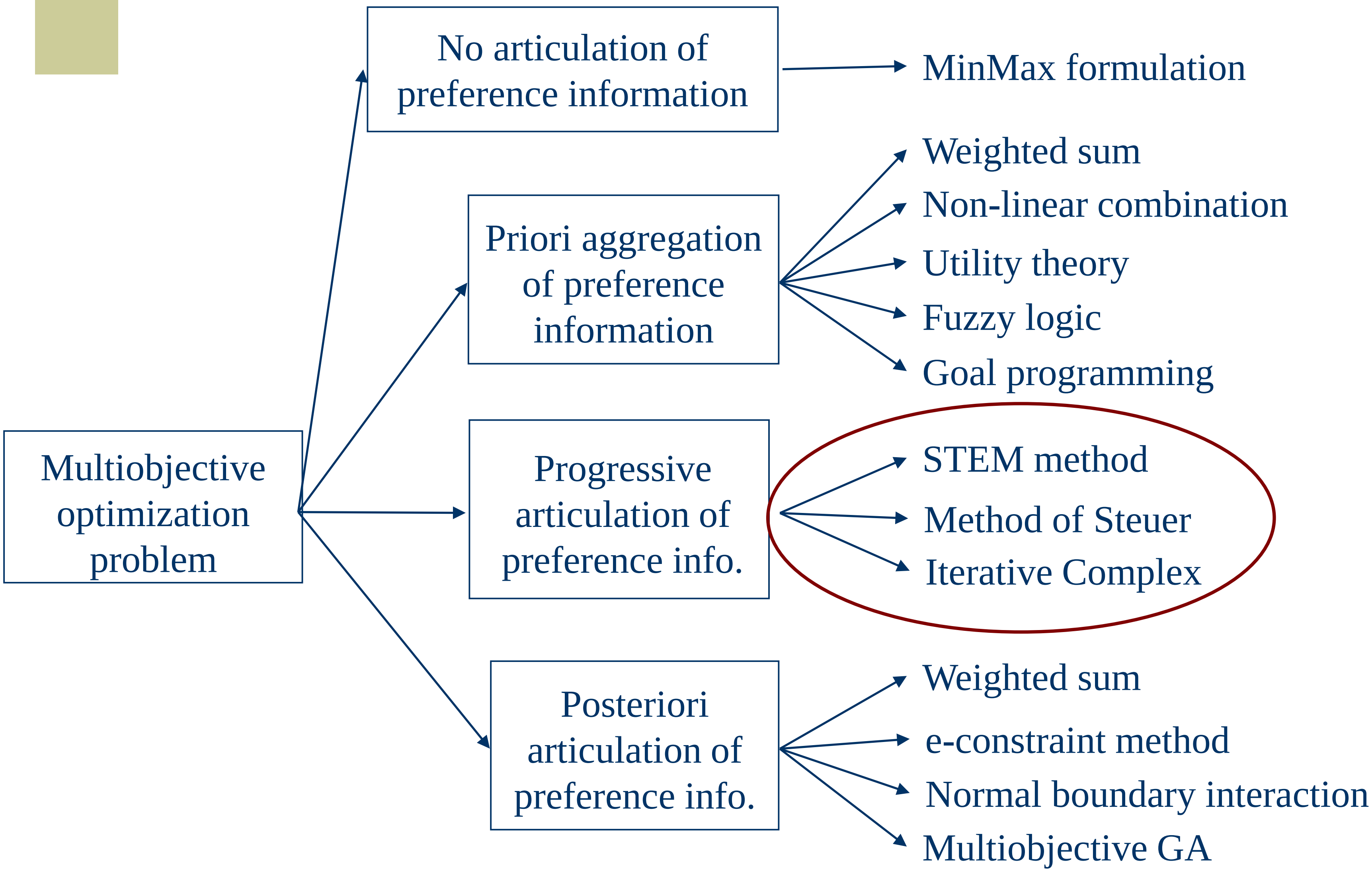
The criteria could be formulated in the following ways:

- Greater than or equal to
- Less than or equal to
- Equal to
- In the range of

Goal programming: *formulation*

- Usually there are no point that fulfils all goals.
- Find the point that "best" matches the "goals".
- The problem is formulated as to find the point with the shortest distance to the "utopian point"/goal area.
  - Use some sort of L_p norm to measure the distance.
  - Lexicographic approach.

Articulation of preference info.



Iterative methods: *STEM*

The Problem is formulated as a weighted sum where the solution space is successively reduced.

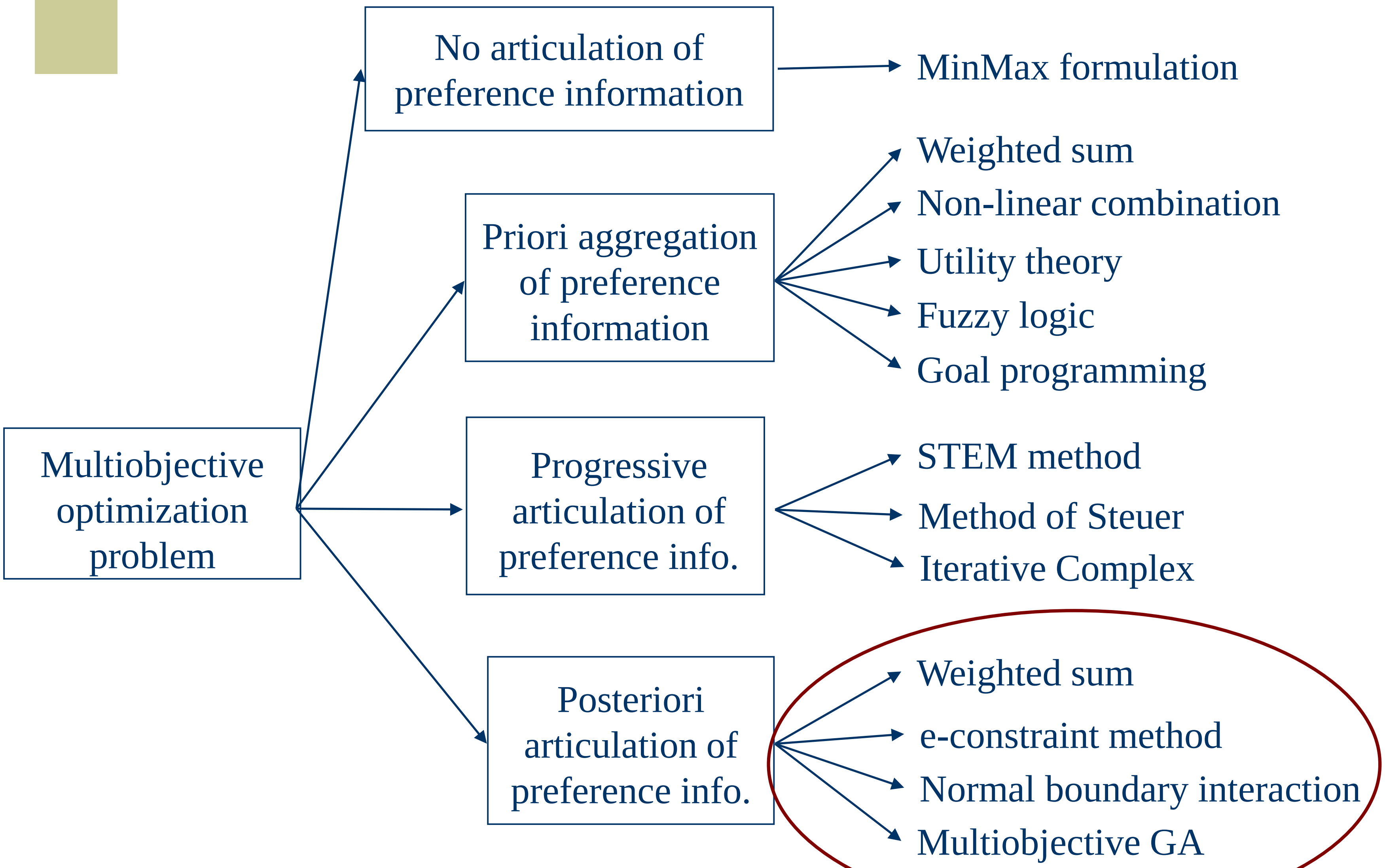
- Find an initial solution $\tilde{f}$.
- Compare with utopian solution F^*.
- If not acceptable determine a relaxation (Δf_j) in one objective.
- Insert $f_j \le \tilde{f}_j + \Delta f_j$ as a new constraint.
- Solve the new problem.
- Satisfied?

Iterative methods: *Iterative Complex*

- Let the decision-maker (DM) determine which is the worst solution.
- Reflect this solution through the centeriod as in the normal Complex.
- Again let the DM point out the worst point

- Hard to do for many parameters
- The DM could take many aspects into account, that could not be mathematically expressed.

Articulation of preference info.



Multiple run approaches

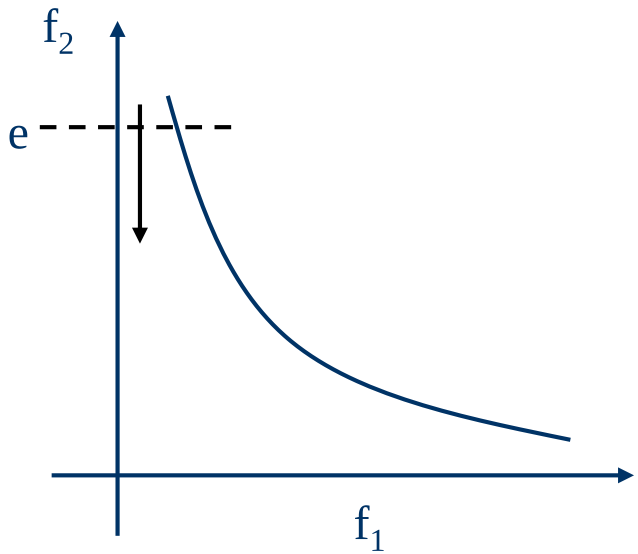
- Weighted sum
- e-constraint method
- Normal boundary interaction

Weighted sum method

- Sample points on the Pareto front by changing the weights of a weighted sum.
- Hard to determine the weights to get an even spread.
- Linear combinations of the objectives can not find points on non-convex Pareto fronts.

e-constraint method

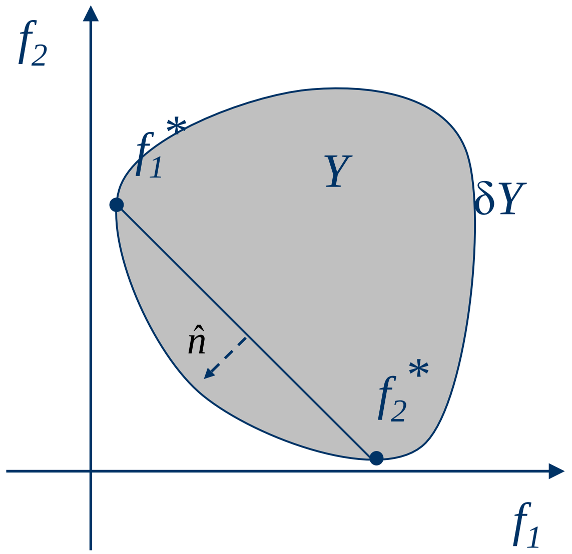
- Sample points on the Pareto front by successively adding a constraint *e* on one of the objectives.



$$\min \quad f_1(\mathbf{x})$$
$$\text{s.t.} \quad \mathbf{x} \in \mathrm{S}$$
$$\mathrm{f}_2 \leq e$$

Normal boundary interaction

Move as far as possible in the direction of the normal of the line that interconnects the individual minima



Multi-objective genetic algorithms

- Tries to spread the population evenly on the Pareto front as the GA evolves.
- Identify the Pareto front in one optimization run.

Example:

Selection based:

Vector evaluating GA (VEGA)

Niched Pareto GA (NPGA)

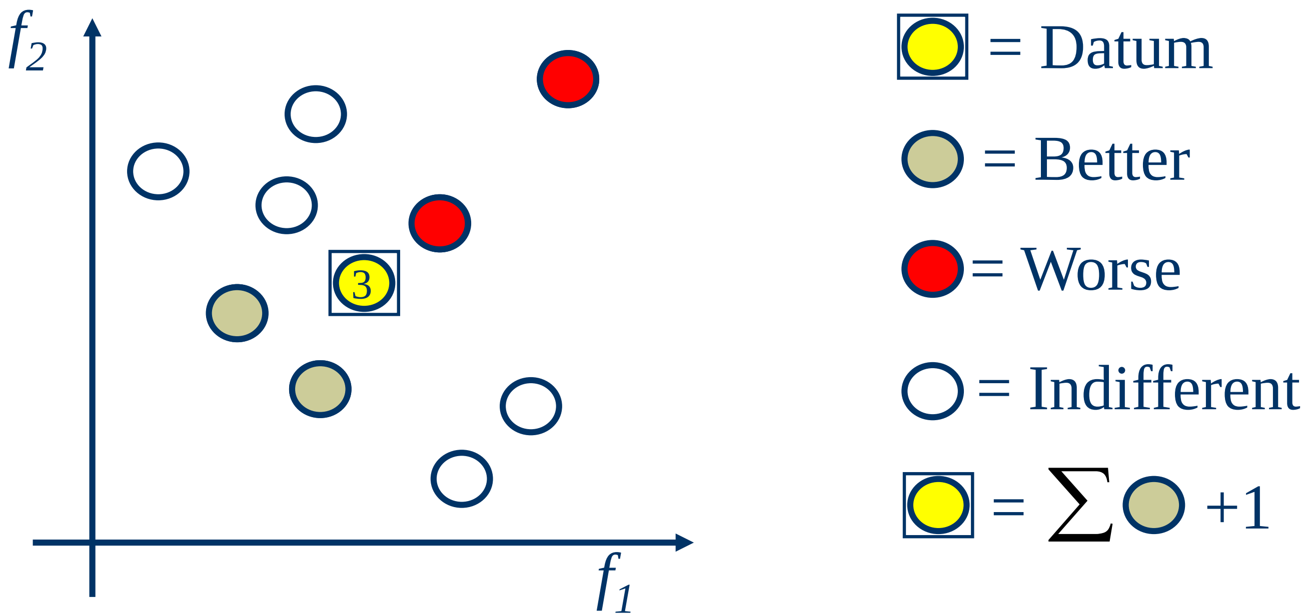
Ranking based:

Non-dominated sorting GA (NSGA)

Multi-objective GA (MOGA)

Multi-Objective GA

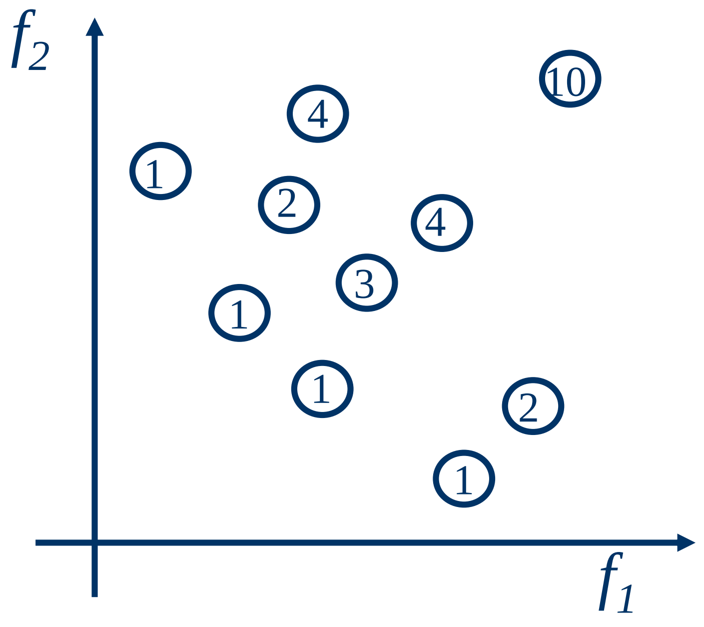
Use Pareto dominance to rank the population



(Fonseca and Flemming, 1995)

MOGA

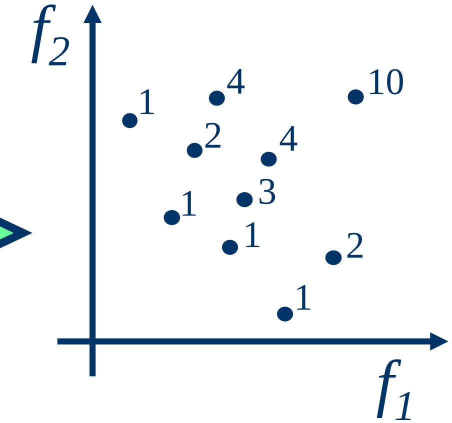
Use Pareto dominance to rank the population

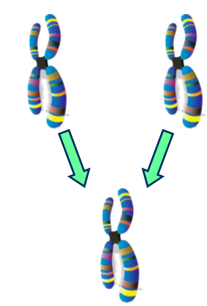


(Fonseca and Flemming, 1995)

MO-Struggle Algorithm

- Initialize the population
- Rank population according to Pareto dominance
- Select parents
- Perform crossover and mutation
- Find the individual <u>most similar</u> to the child,replace it if the child is better.

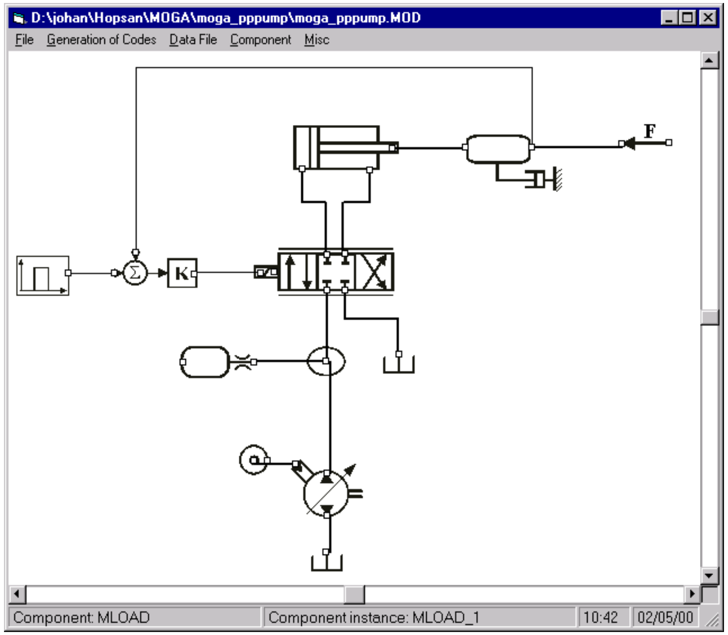




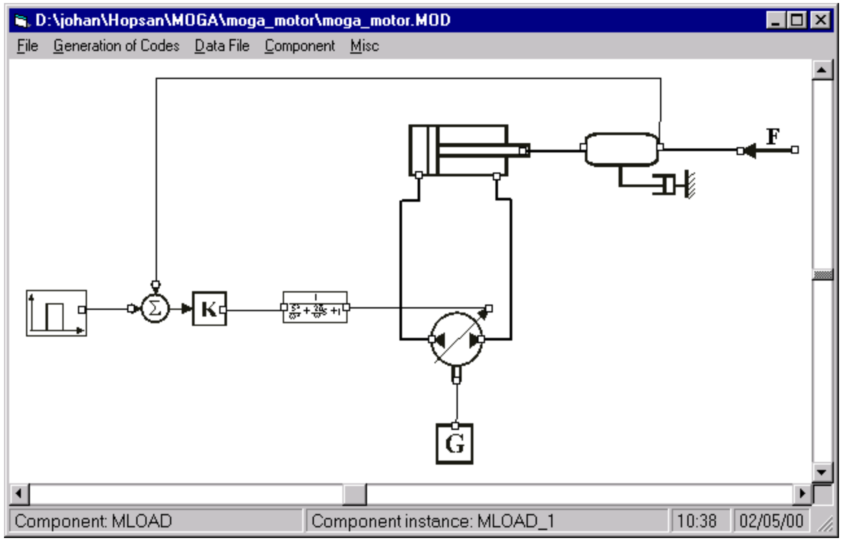
<u>Similarity measures</u>

Euclidean distance in attribute or parameter space, or a mix of both.

Application example

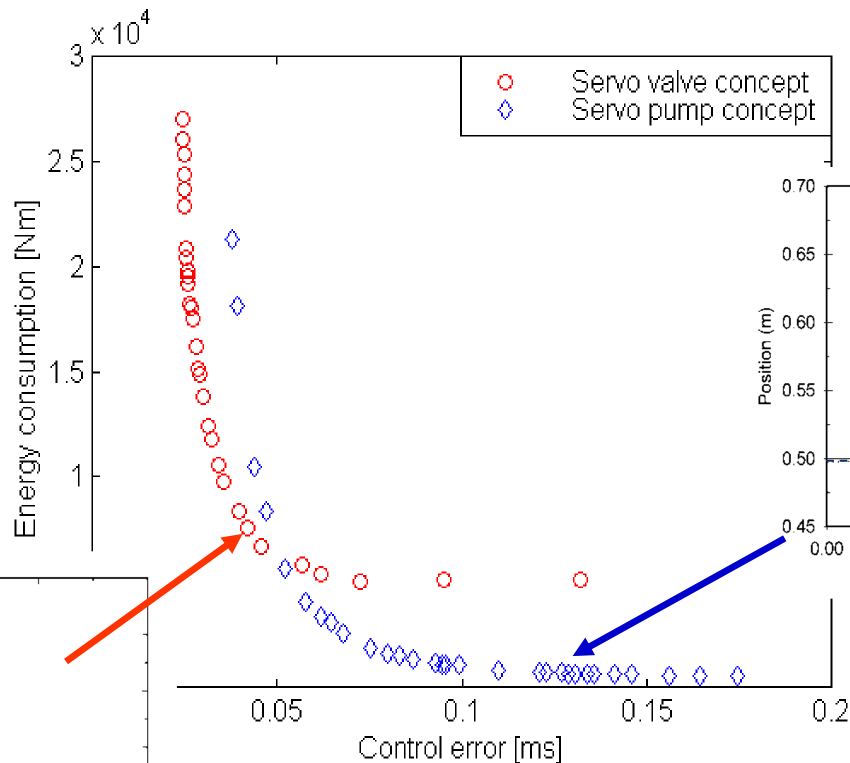


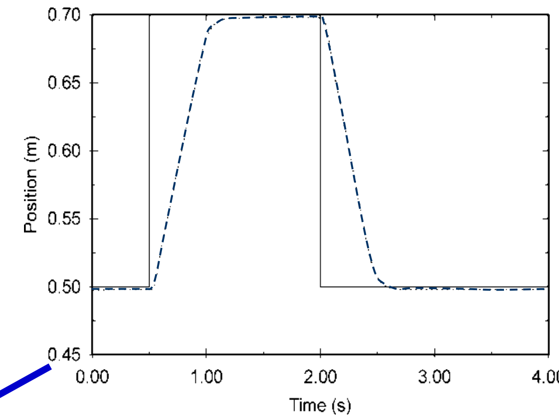
Servo valve system



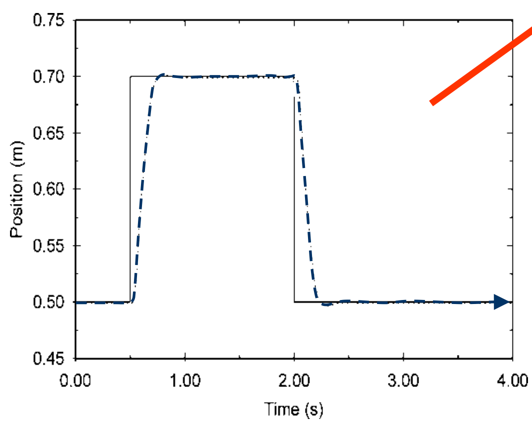
Servo pump system

Application example





Slow pulse response

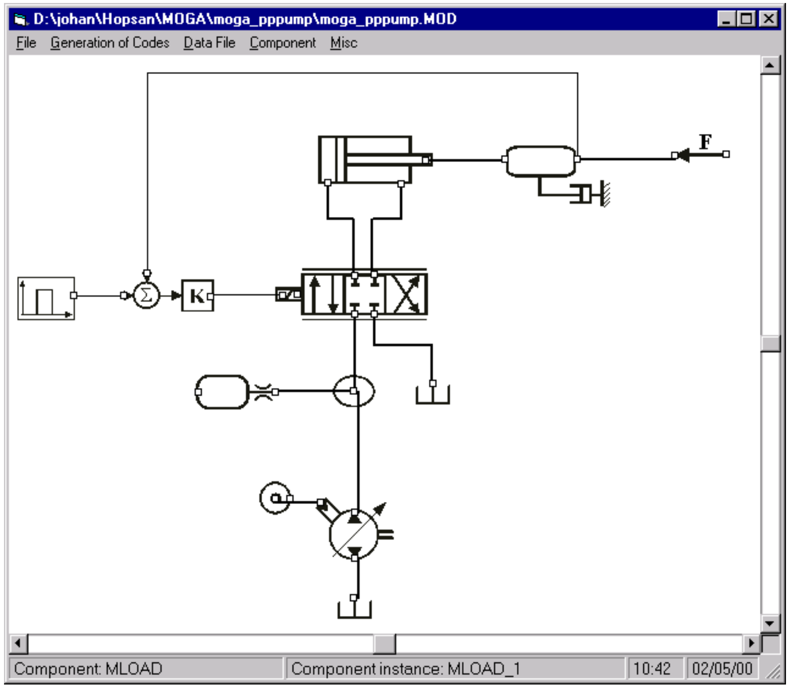


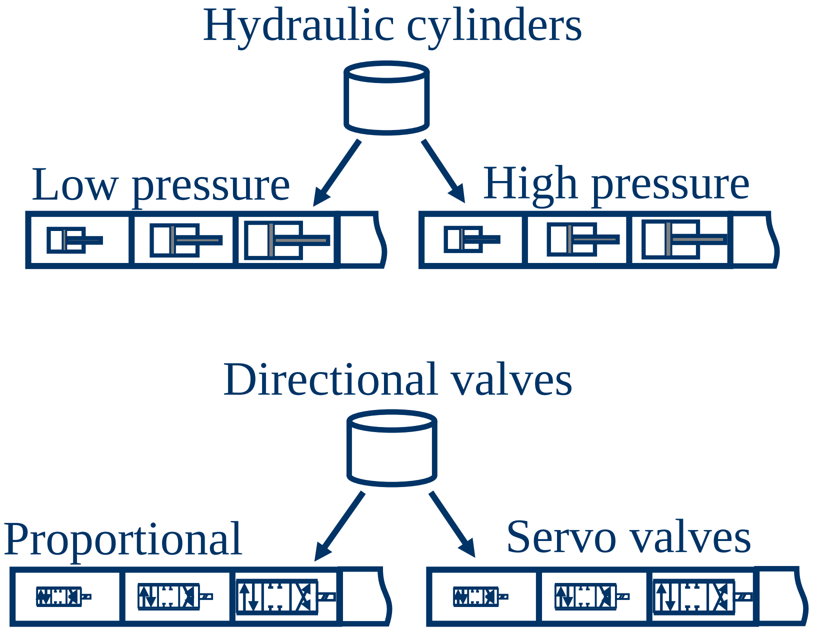
Pareto front showing the trade-off between control error and energy consumption

Fast pulse response

Discrete optimization

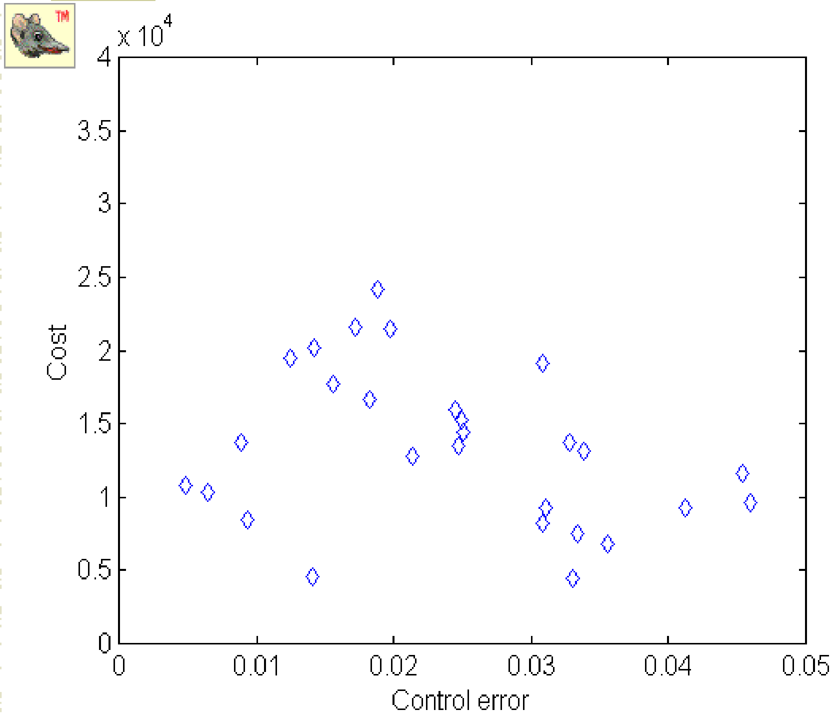
Component databases



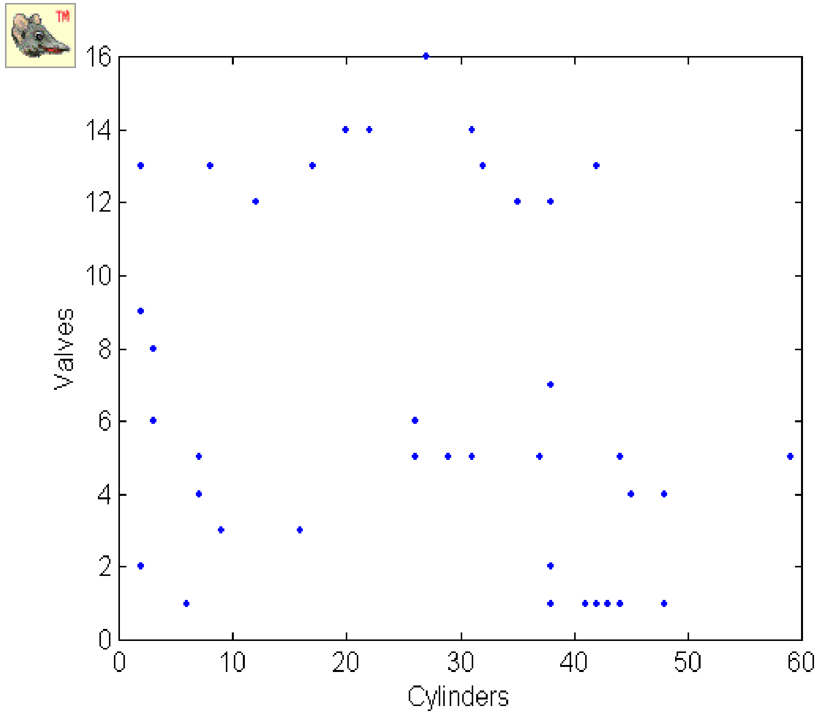


Valve system

The optimization progress

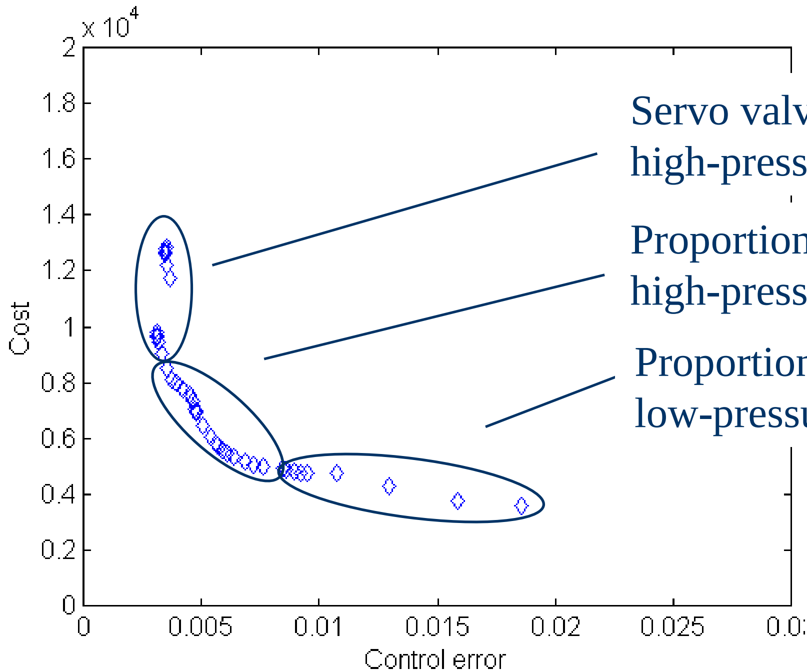


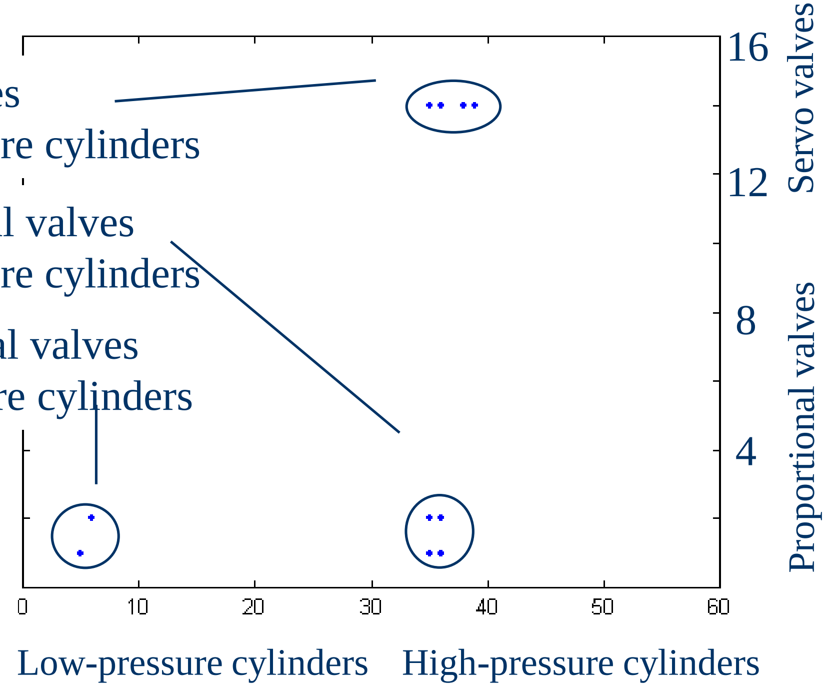
The population in objective space



The population in parameter space: Valves and Cylinders

Optimization result





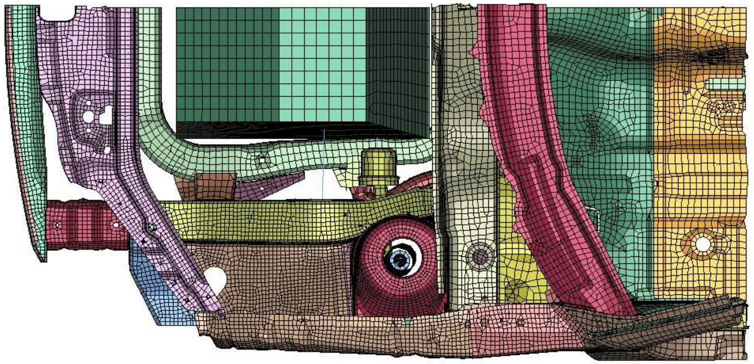
Pareto front

Valve and cylinder selection

The design problem

- **The model:** Saab 9^5, LS Dyna, 56000 shell elements
- **Impact:** rigid wall, 15.64 m/s
- **CPU time:** 10h, or 3h on a Linux cluster with 4 CPU:s
- **Objectives:** minimize the maximum acceleration

  minimize intrusion



Density	8673.6 kg/m^3
Youngs modulus	$2.06e^{10}$ N/m^2
Poisons ratio	0.3
Yield stress	$3.8e^{8}$ N/m^2

The response surface

- **The model –** quadratic response surface

$$y = b_0 + \sum_{i=1}^{n} b_i x_i + \sum_{i=1}^{n} b_{ii} x_{ii}^2 + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} b_{ij} x_i x_j$$

- **Design setup** – D-optimal criterion

  25 function evolutions
  5 design parameters

- **Model accuracy RMS**$_{\text{norm}}$– acceleration 5.8%
  intrusion 4.1%

$$RMS_{err} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(y_i - \hat{y}_i \right)^2}$$

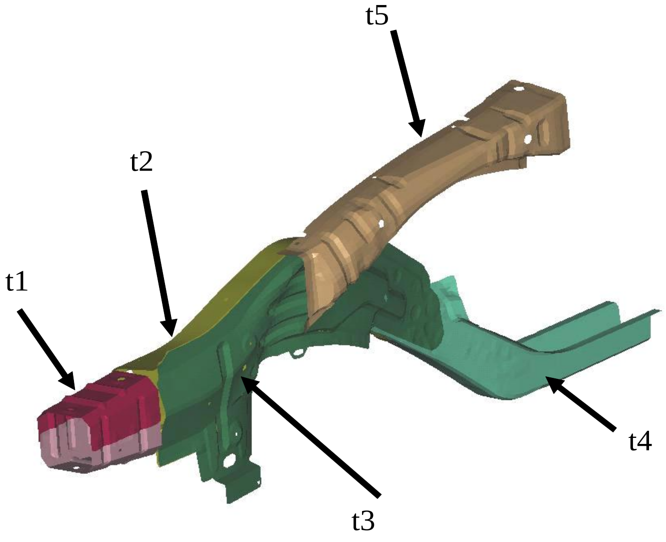
$$RMS_{norm} = \frac{RMS_{err}}{\bar{y}}, \quad \text{where } \bar{y} = \frac{y_{\max} - y_{\min}}{2}$$

Optimization problem

$$\min \mathbf{F}(\mathbf{x}) = \left(f_1(t), f_2(t) \right)^T$$
$$s.t.\ \ t \in S$$
$$t = (t_1, t_2, \ldots, t_5)^T$$

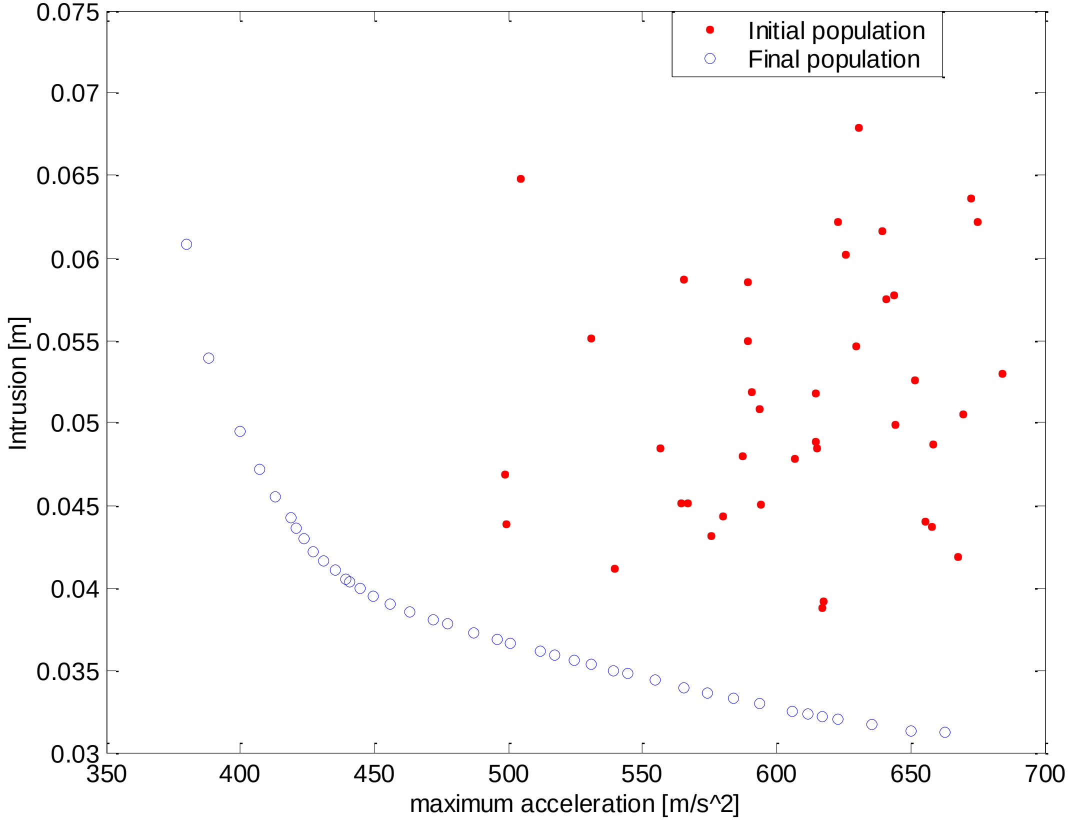
$$f_1(\mathbf{t}) = \text{max acceleration}$$

$$f_2(\mathbf{t}) = \text{intrusion}$$



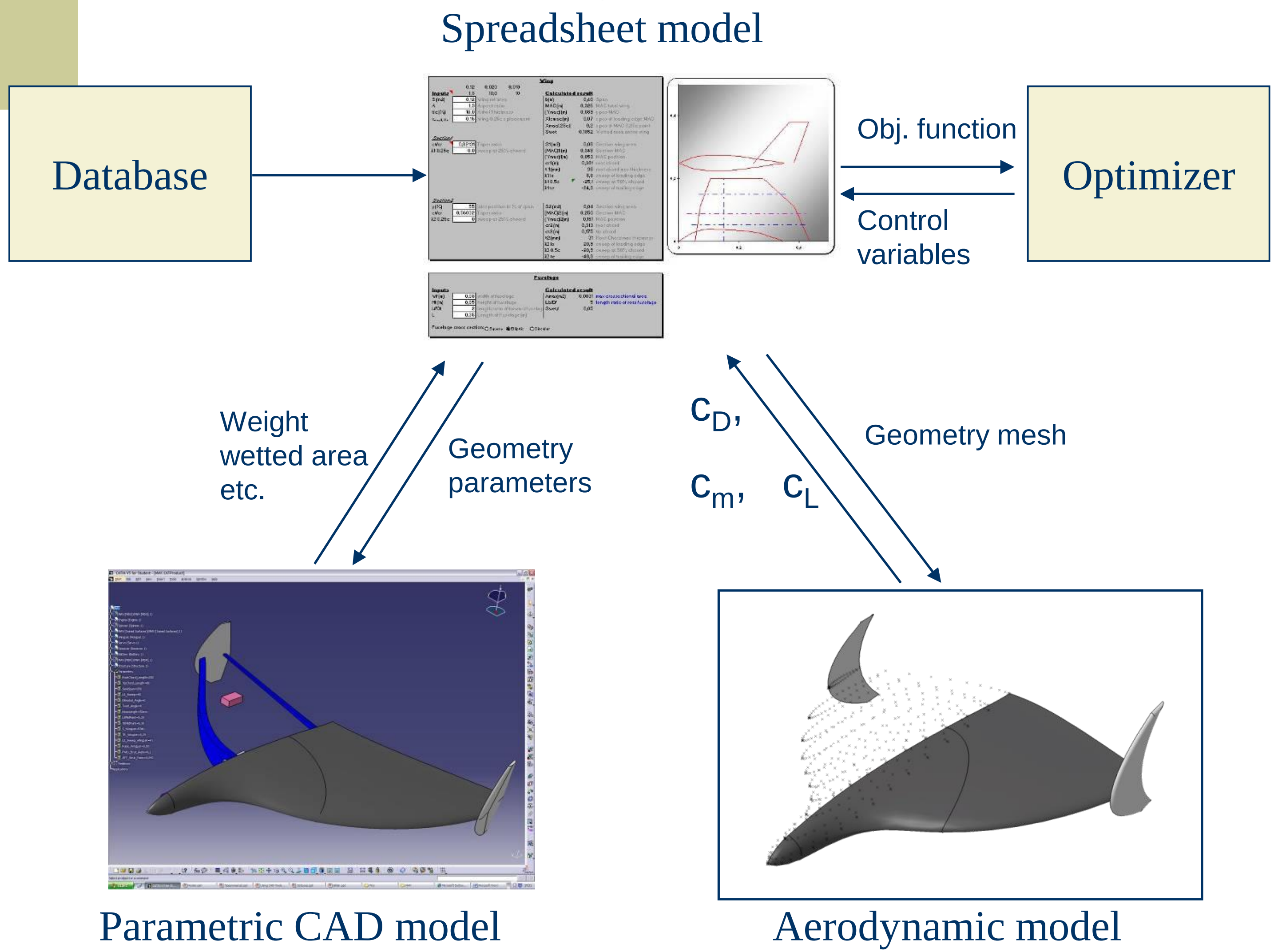
Crash box	t1 = 1.50 ±0.5 mm
Mid rail c-profile	t2 = 1.65 ±0.5 mm
Mid rail closing plate	t3 = 1.65 ±0.5 mm
Rail extension	t4 = 1.95 ±0.5 mm
Upper rail	t5=1.20 ±0.5 mm

Results: *Pareto optimal front*

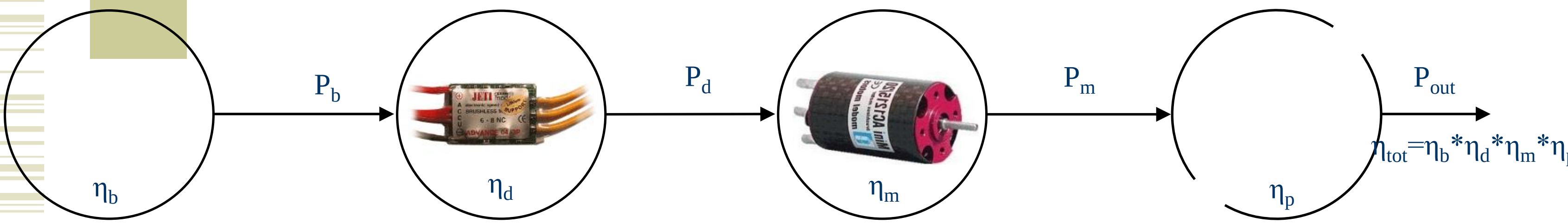


Pareto front visualizing the trade-off between maximum acceleration and intrusion into the passenger compartment

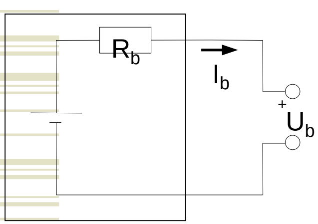
MAV - Design Framework



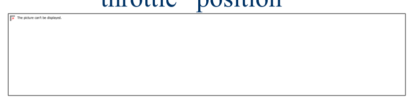
Modeling – Propulsion System

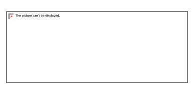


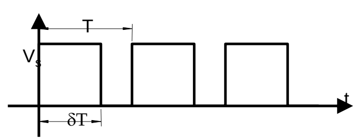
- **Battery**
  - ➢ Cell resistance
  - ➢ Cell capacity
  - ➢ Cell voltage
  - ➢ Nr. of serial cells
  - ➢ Nr. of parallel cells



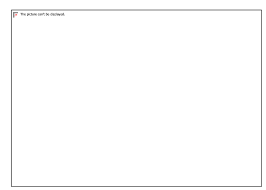
- **Controller**
  - ➢ Resistive losses
  - ➢ Losses depending on “throttle” position

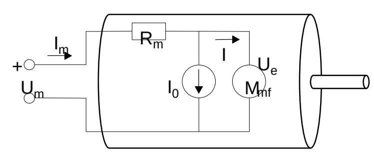






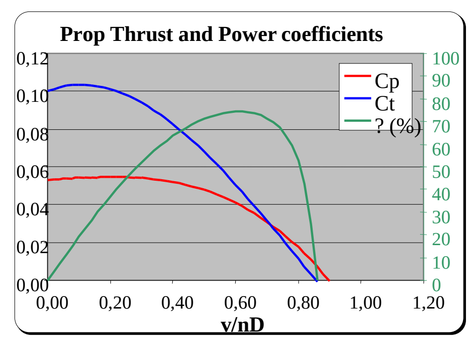
- **Classical electric motor model**
  - ➢ K_v, I_0, R_m



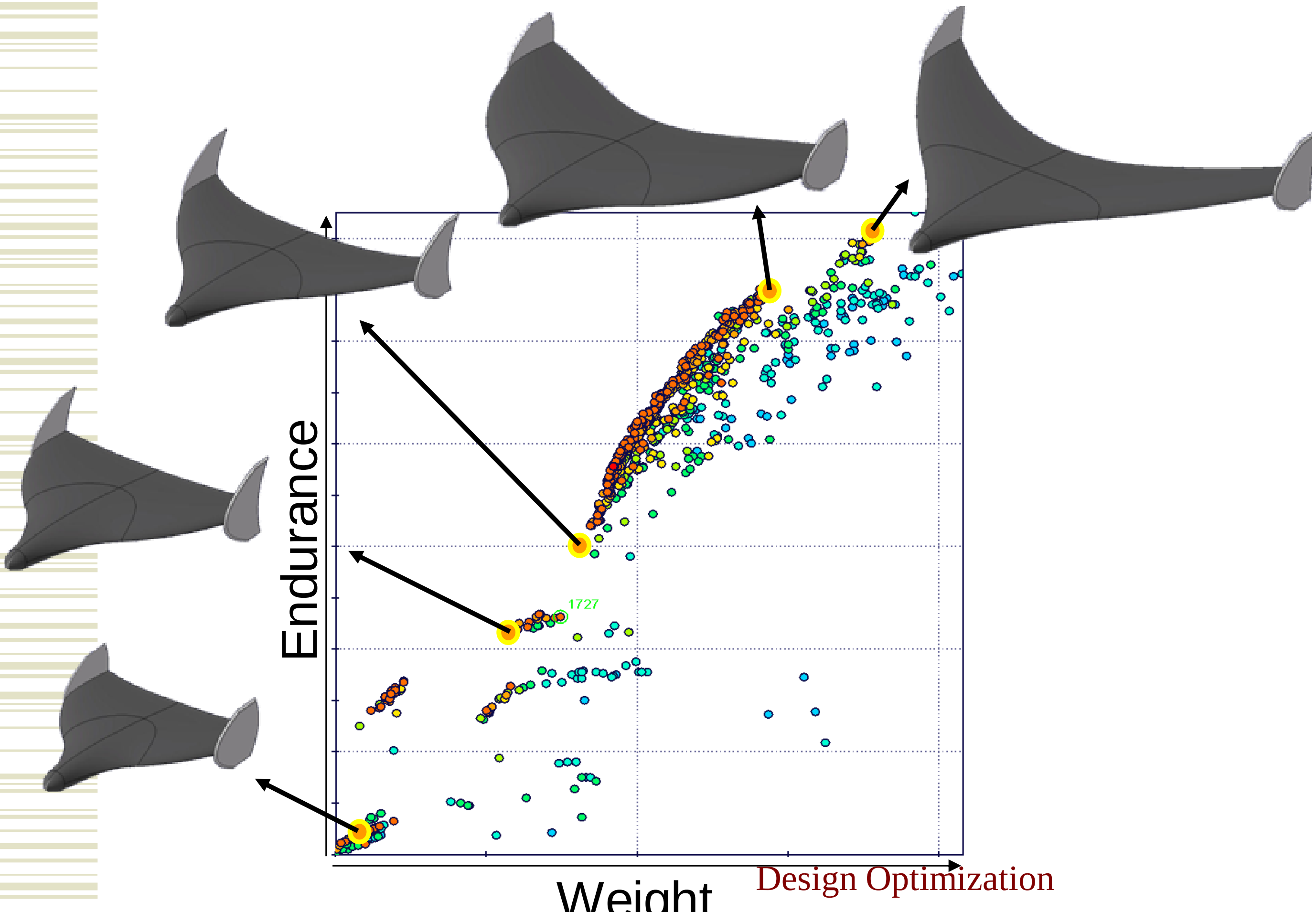


- **Blade element method**

Performance characterized by Thrust and Power coefficients as function of advance ratio v/nD

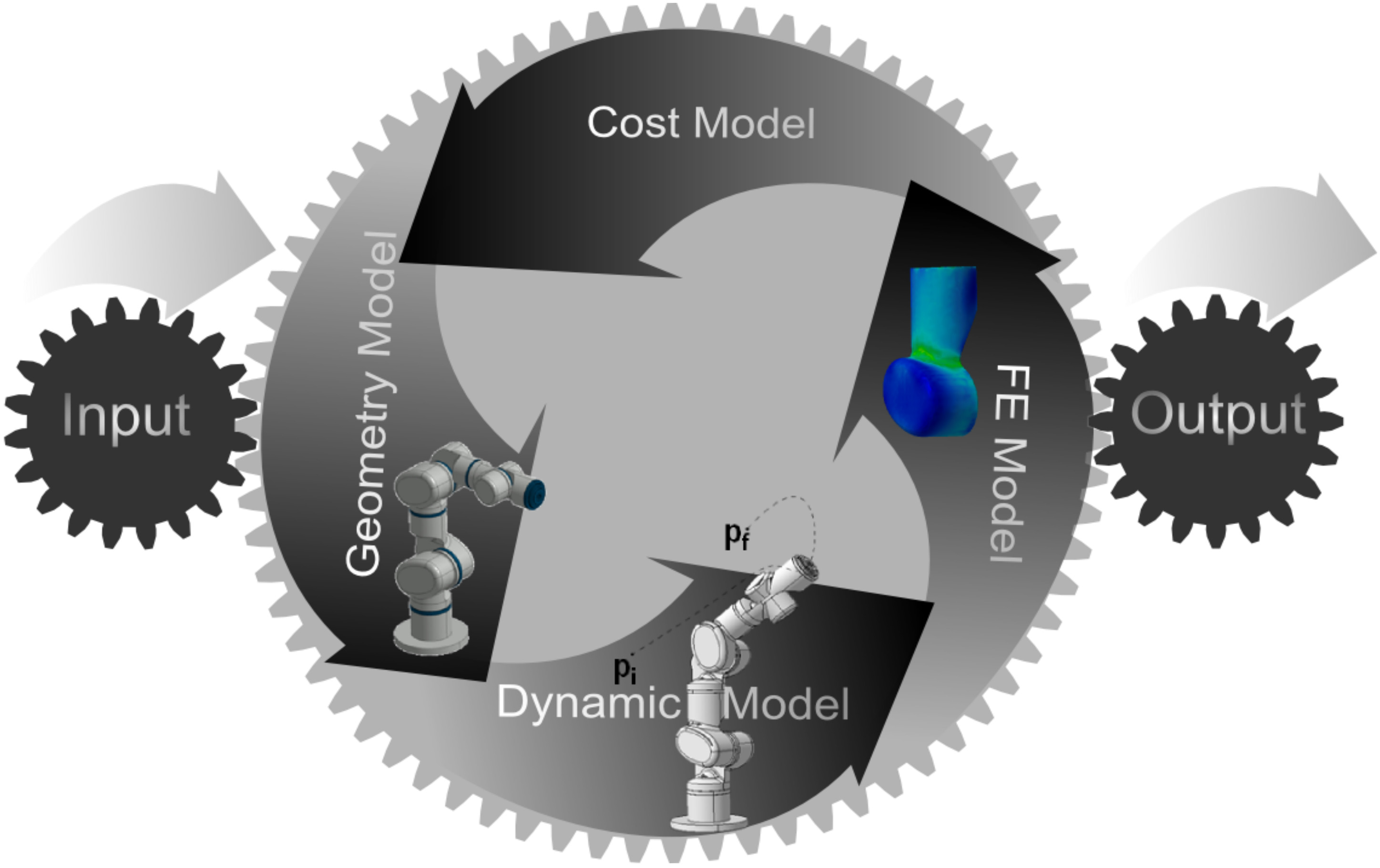


Closing the Loop - MAV Prototyping



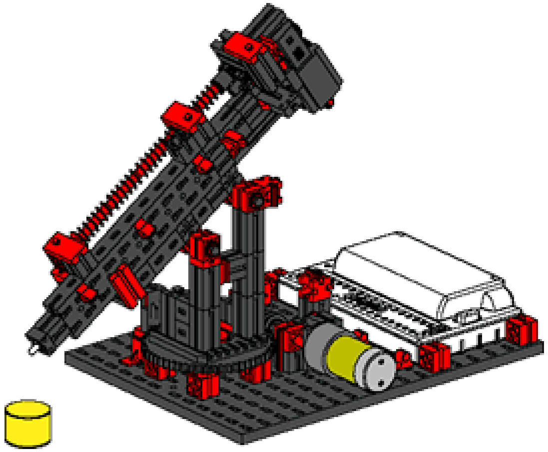
Design Automation to enable MDO

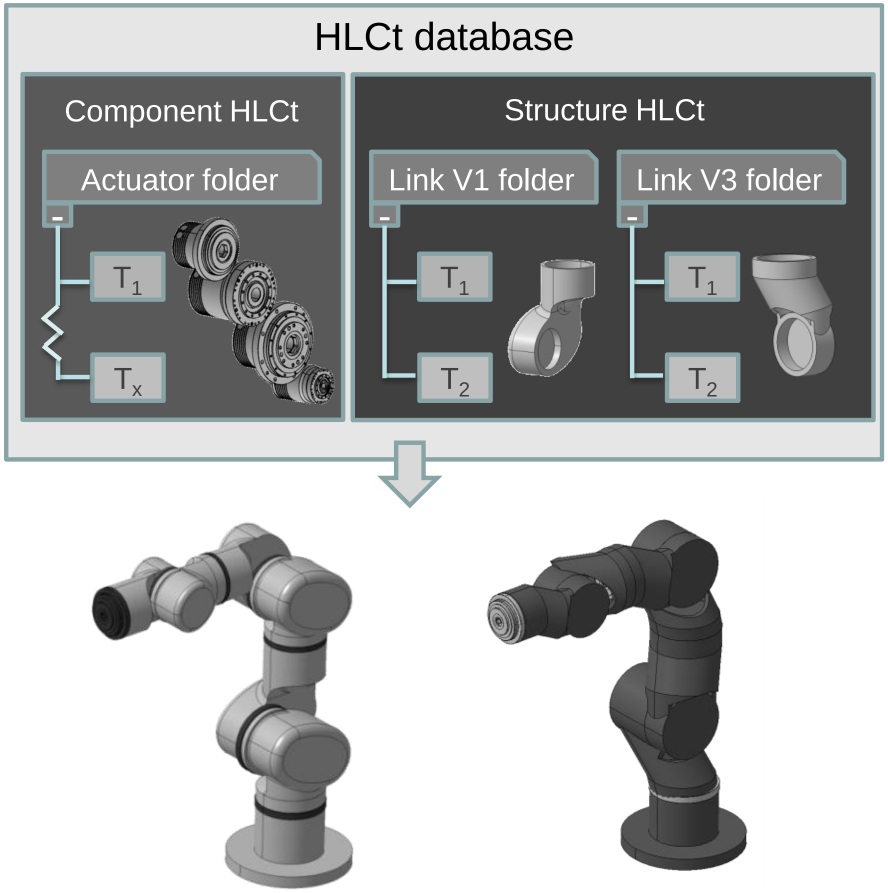
- 1. Which design tools and models are required?



High Level CAD template -> Virtual LEGO

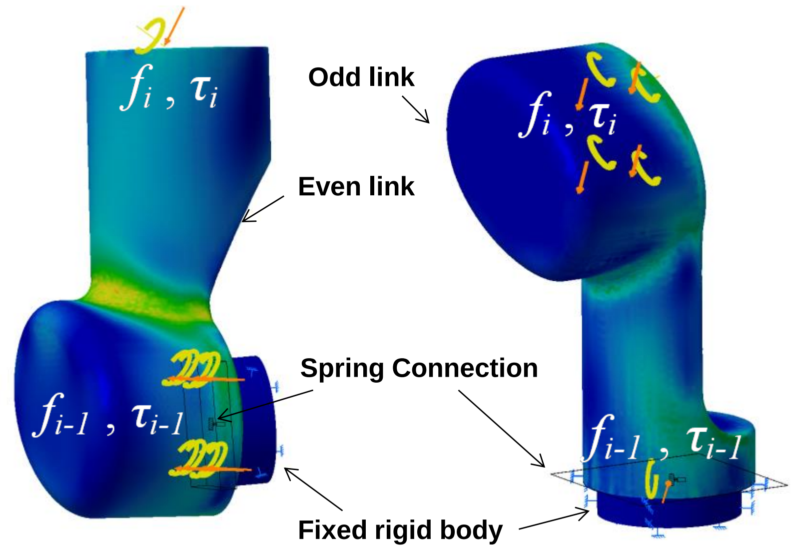
- 2. How to generate the geometry parametrically?

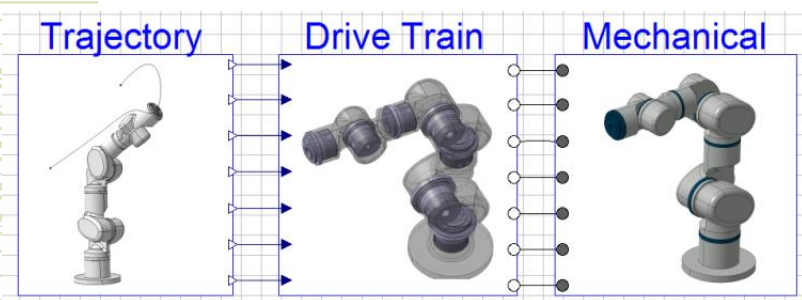




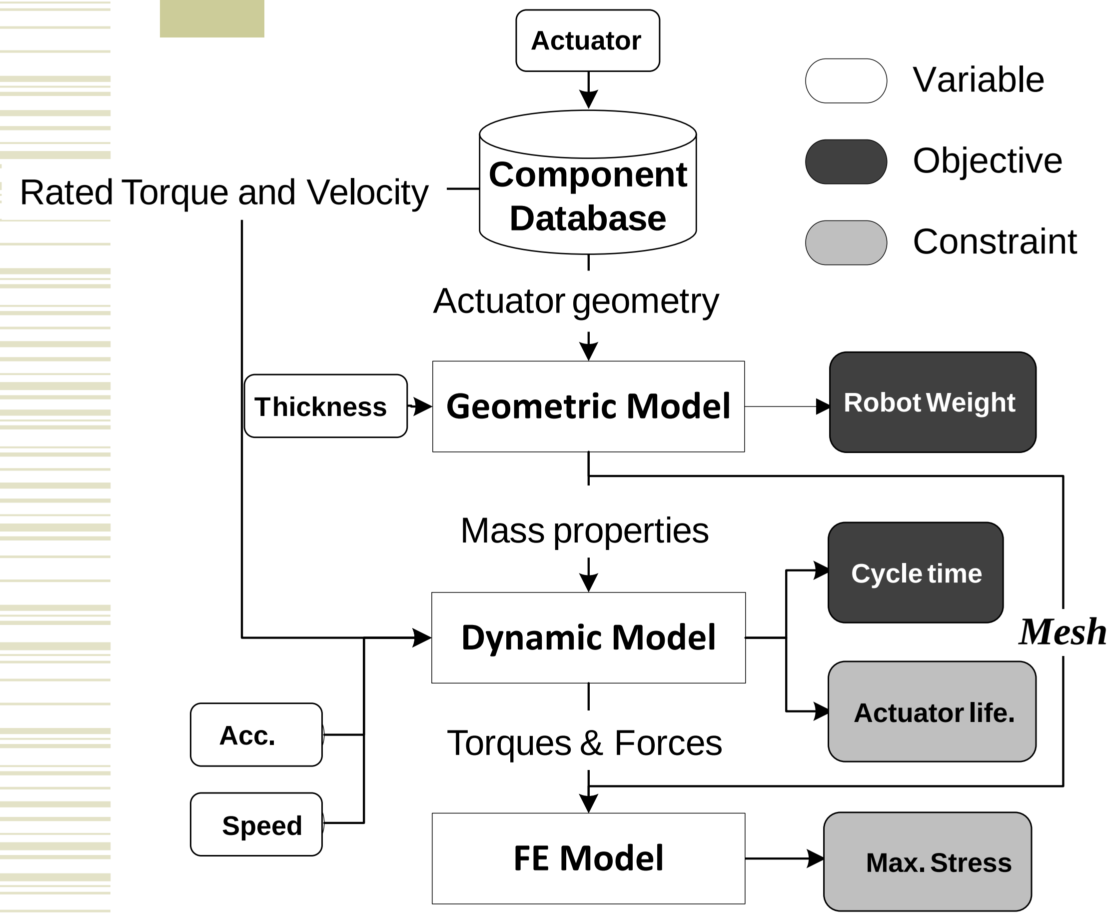
Integrated Design

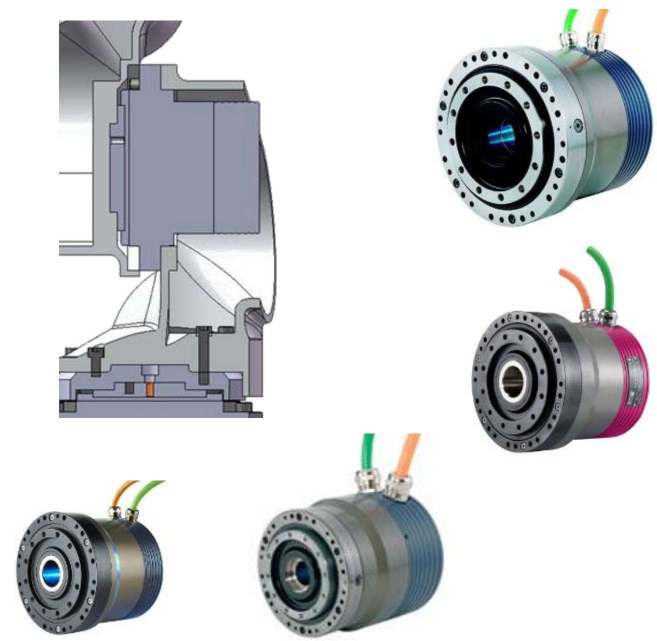
•3. How to achieve design integration?



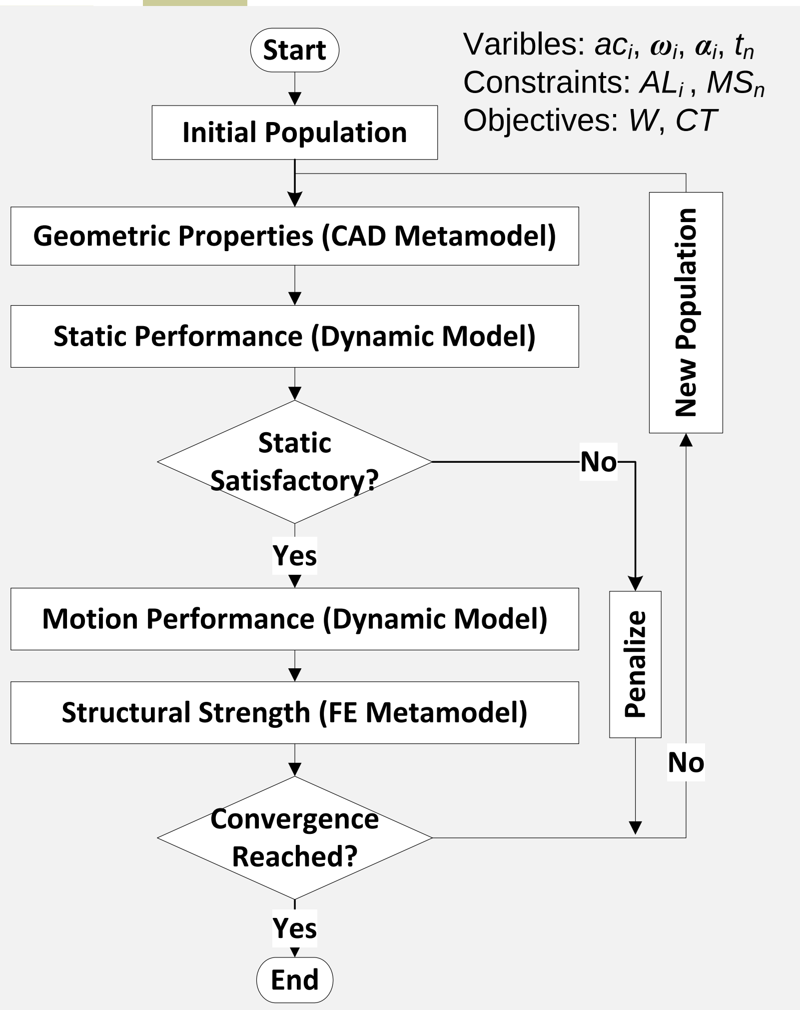


Multidisciplinary Optimization Framework

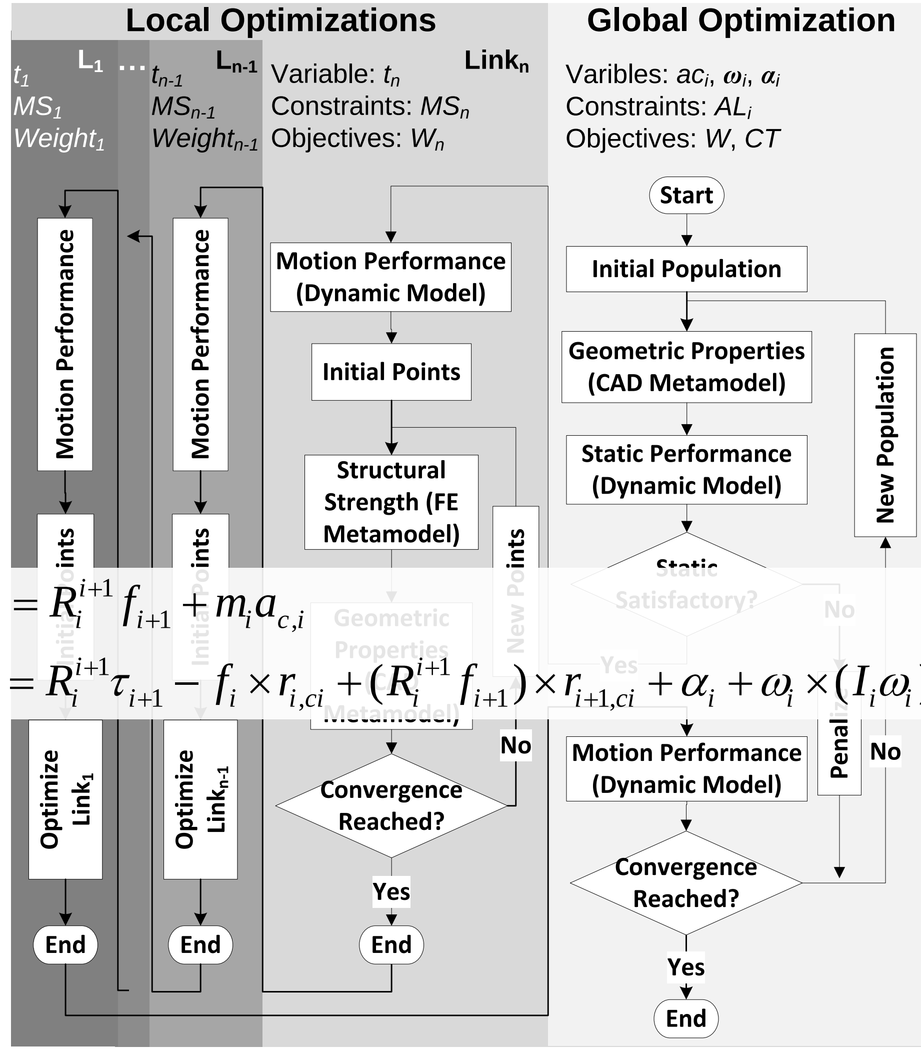




Multi Level Optimization



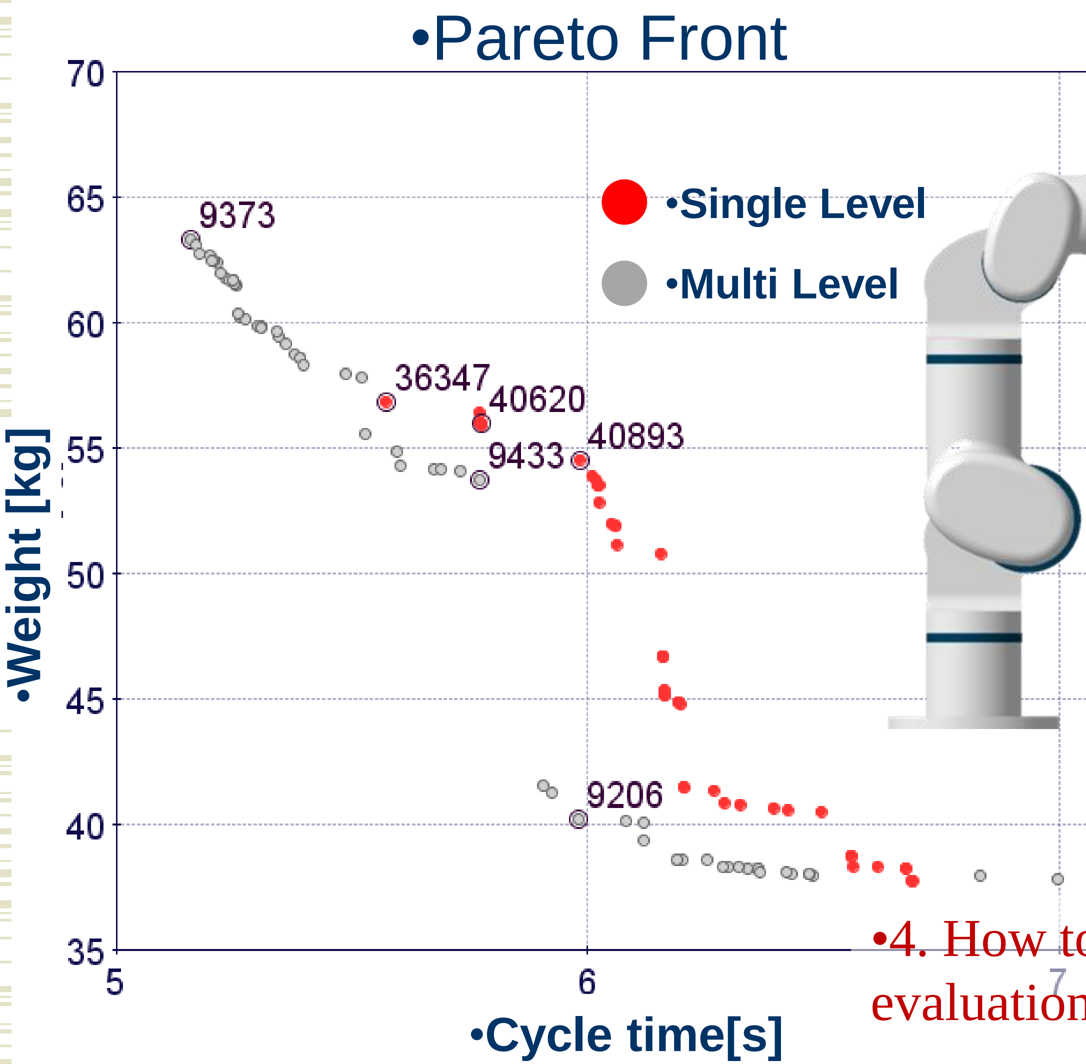
•**Multi Level Optimization**

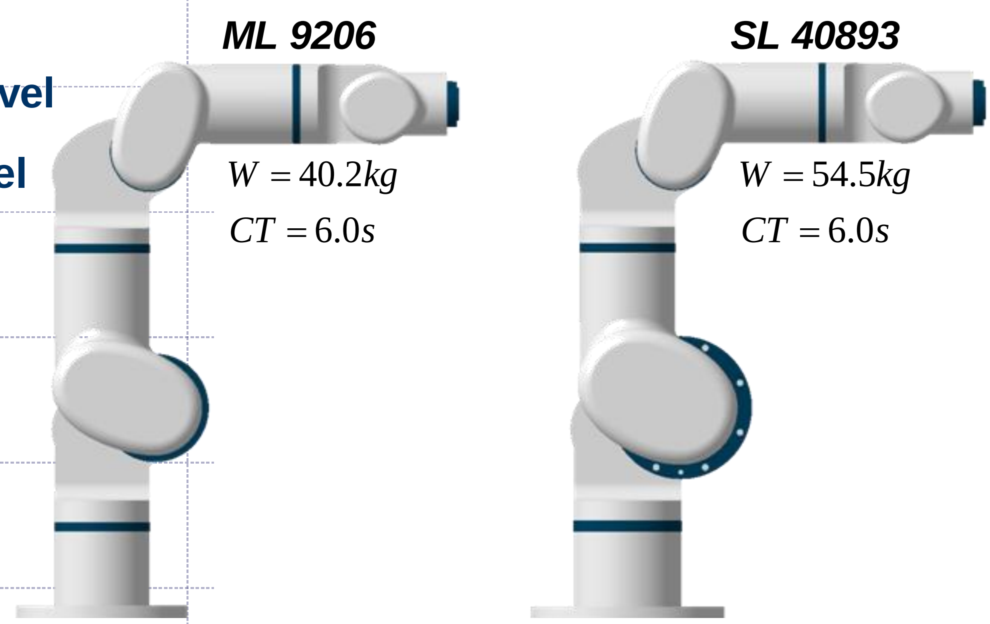


$$f_i = R_i^{i+1} f_{i+1} + m_i a_{c,i}$$

$$\tau_i = R_i^{i+1} \tau_{i+1} - f_i \times r_{i,ci} + (R_i^{i+1} f_{i+1}) \times r_{i+1,ci} + \alpha_i + \omega_i \times (I_i \omega_i)$$

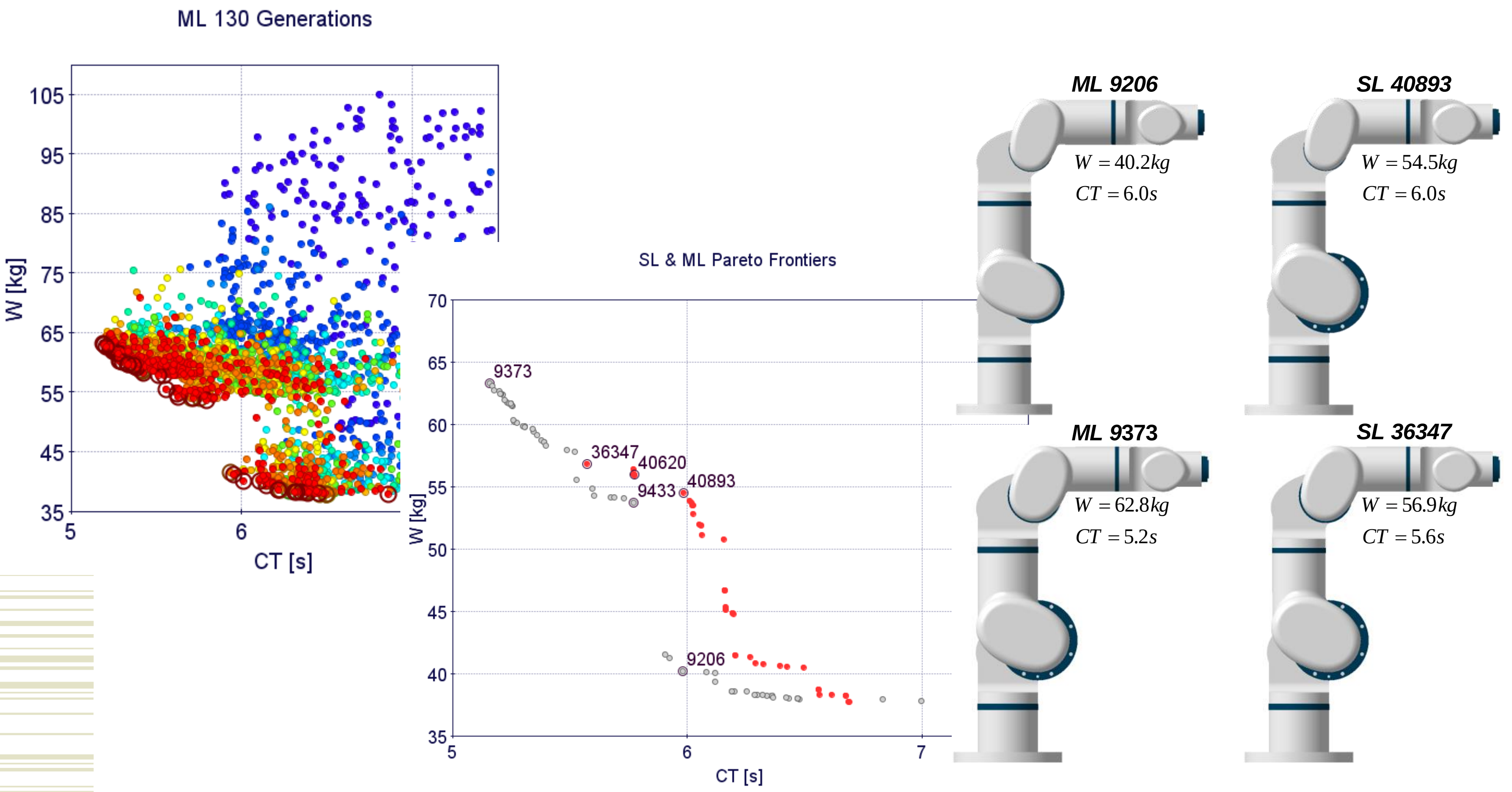
Single Robot Optimization





•4. How to achieve time efficient concept evaluations?

Research project: Industrial robot design



Industrial Robot Family Optimization