Trade-Offs, Strategies and Negotiation in Engineering Design

Erik K. Antonsson, Ph.D., P.E.
Professor of Mechanical Engineering

Engineering Design Research Laboratory
http://www.design.caltech.edu/
Division of Engineering and Applied Science
California Institute of Technology
Pasadena, California

CIMMS Focused Workshop on
Uncertainty Management in Engineering Design

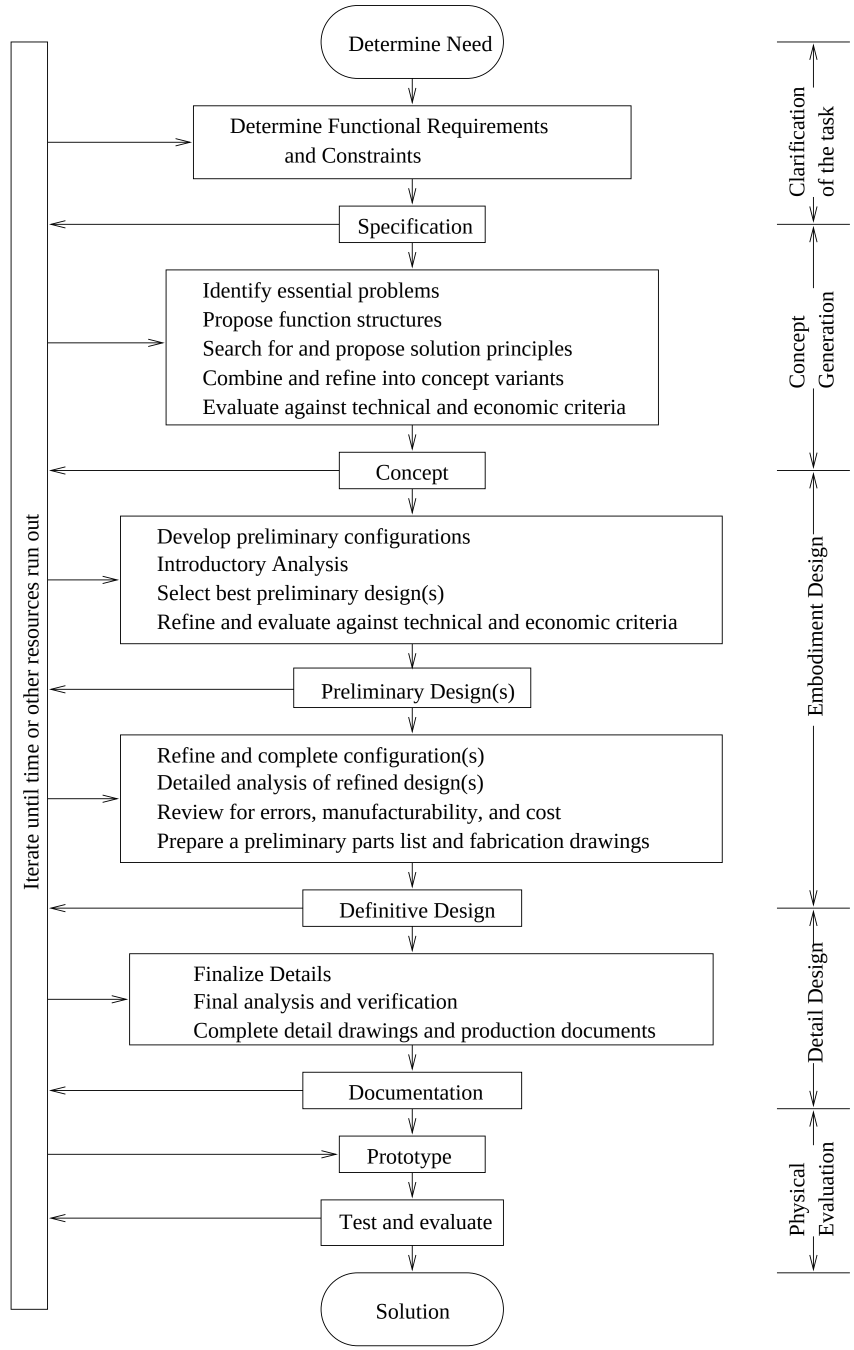
May 23, 2002

Collaborators:

- Michael Scott
- William Law
- Kevin Otto
- Kris Wood

- Ford Motor Company
- Volkswagen, Wolfsburg

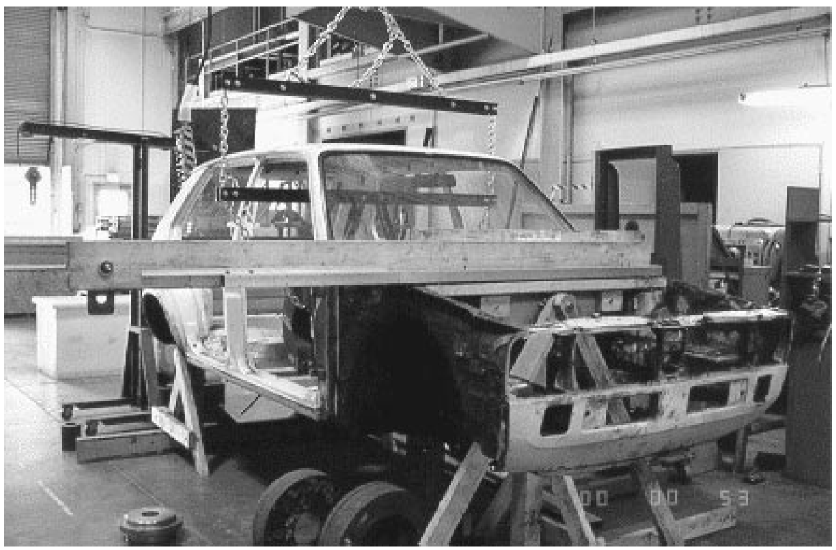
Traditional Engineering Design



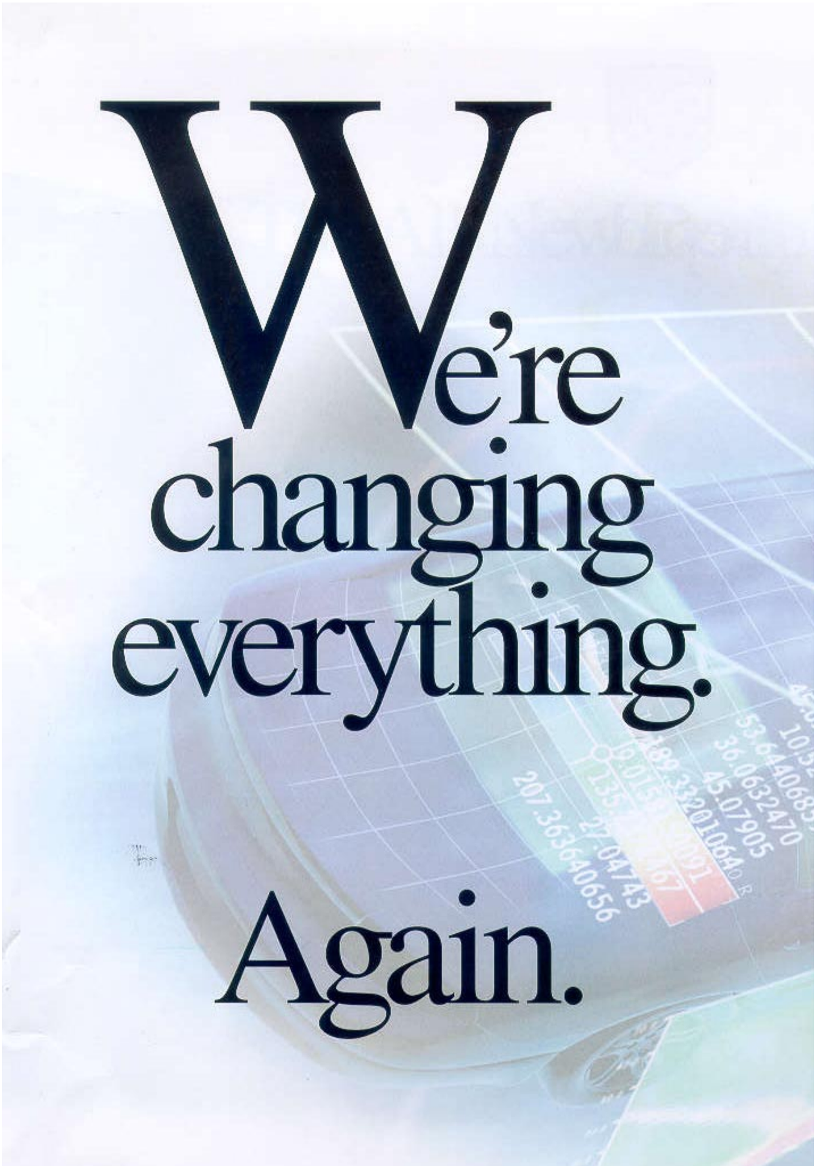
[G. Pahl and W. Beitz, *Engineering Design*, The Design Council, Springer-Verlag, New York, 1984, page 41.]

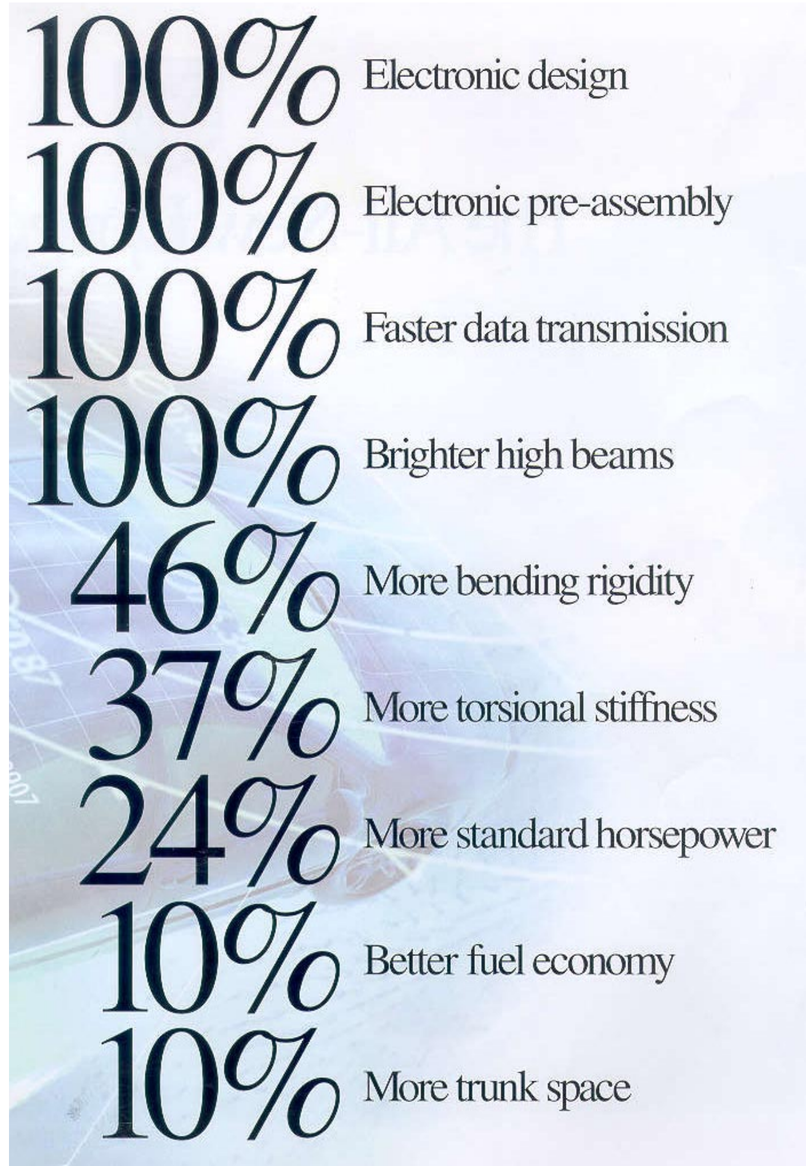
Example: Vehicle Structure Design

The straightforward problem:



	Current	**Targets**
K_B (N/mm)	2480	2730
K_T (N-m/deg)	4920	5420
m (kg)	160	144


We're
changing
everything.
Again.


100% Electronic design
100% Electronic pre-assembly
100% Faster data transmission
100% Brighter high beams
46% More bending rigidity
37% More torsional stiffness
24% More standard horsepower
10% Better fuel economy
10% More trunk space

Load Test Results:

RABBIT CHASSIS PERFORMANCE TEST

TORSIONAL STIFFNESS TEST (WINDSHIELD AND REAR HATCH INTACT)

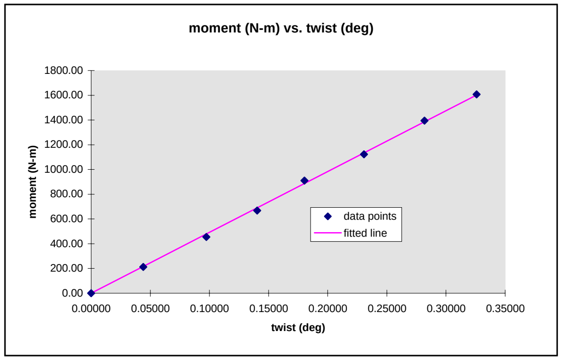
TEST DATE : July 11
indicator radius : 1.42 m (56 in)
moment arm : 1.65 m (65 in)

weight (lbf)	deflection (0.001 in)
0	0
29	43
62	95
91	137
124	176
153	225
190	275
219	318

load (N)	moment (N-m)	deflection (mm)	twist (deg)
0.00	0.00	0.00	0.00000
128.99	212.84	1.09	0.04407
275.78	455.03	2.41	0.09736
404.77	667.87	3.48	0.14041
551.55	910.06	4.47	0.18038
680.54	1122.90	5.72	0.23060
845.12	1394.45	6.99	0.28184
974.11	1607.28	8.08	0.32591

y = mx + c
slope : 4960.74 N-m/deg
y - intercept : -10.16 N-m

y = mx + 0
slope : 4917.04 N-m/deg



RABBIT CHASSIS PERFORMANCE TEST

BENDING STIFFNESS TEST (REAR HATCH AND WINDSHIELD INTACT)

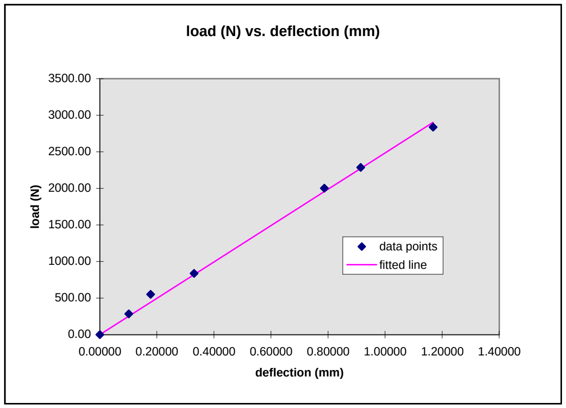
TEST DATE : July 16

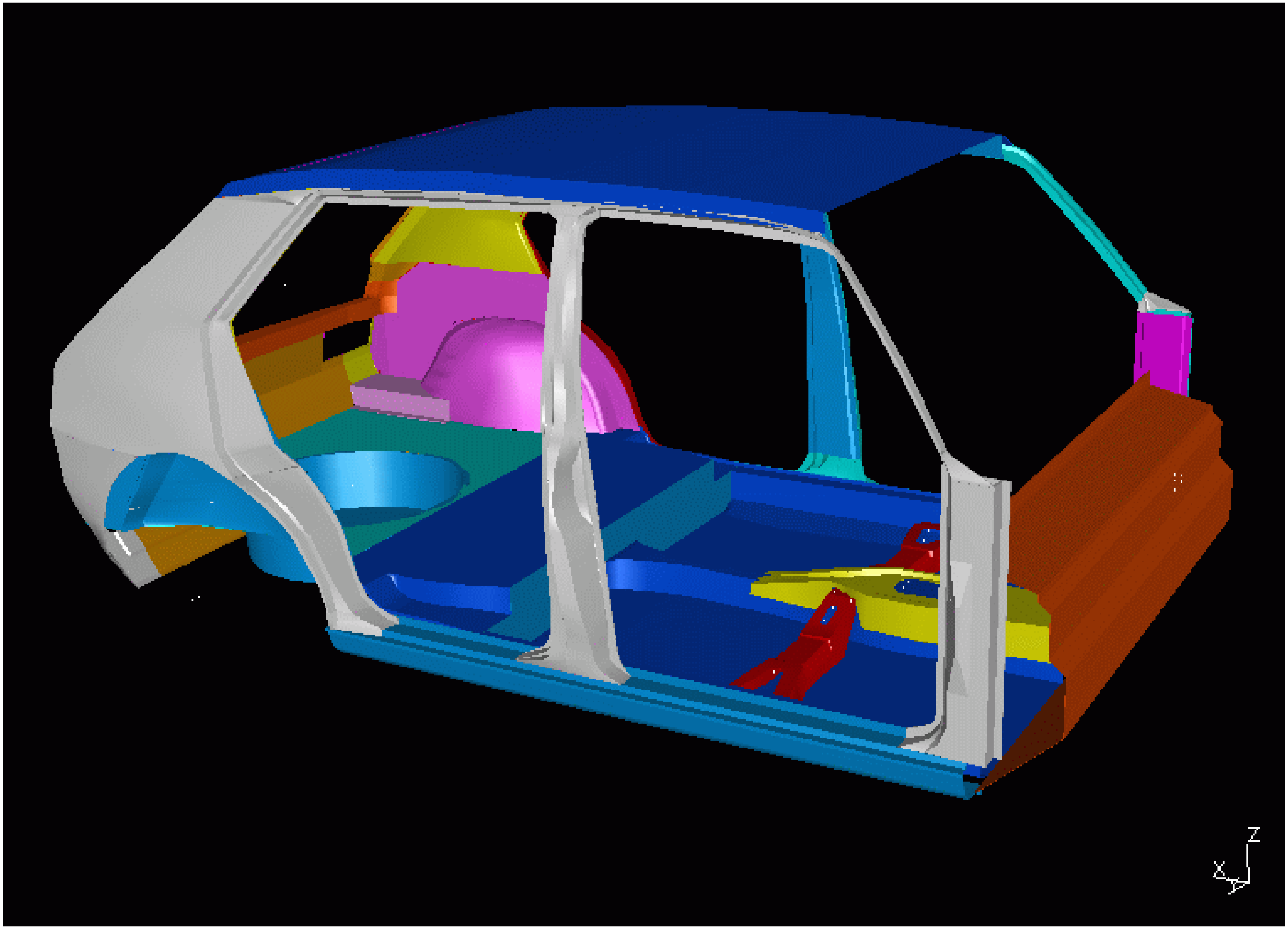
deflection (in)	load (lbf)
0	0
0.004	64
0.007	124
0.013	188
0.031	450
0.036	514
0.046	638

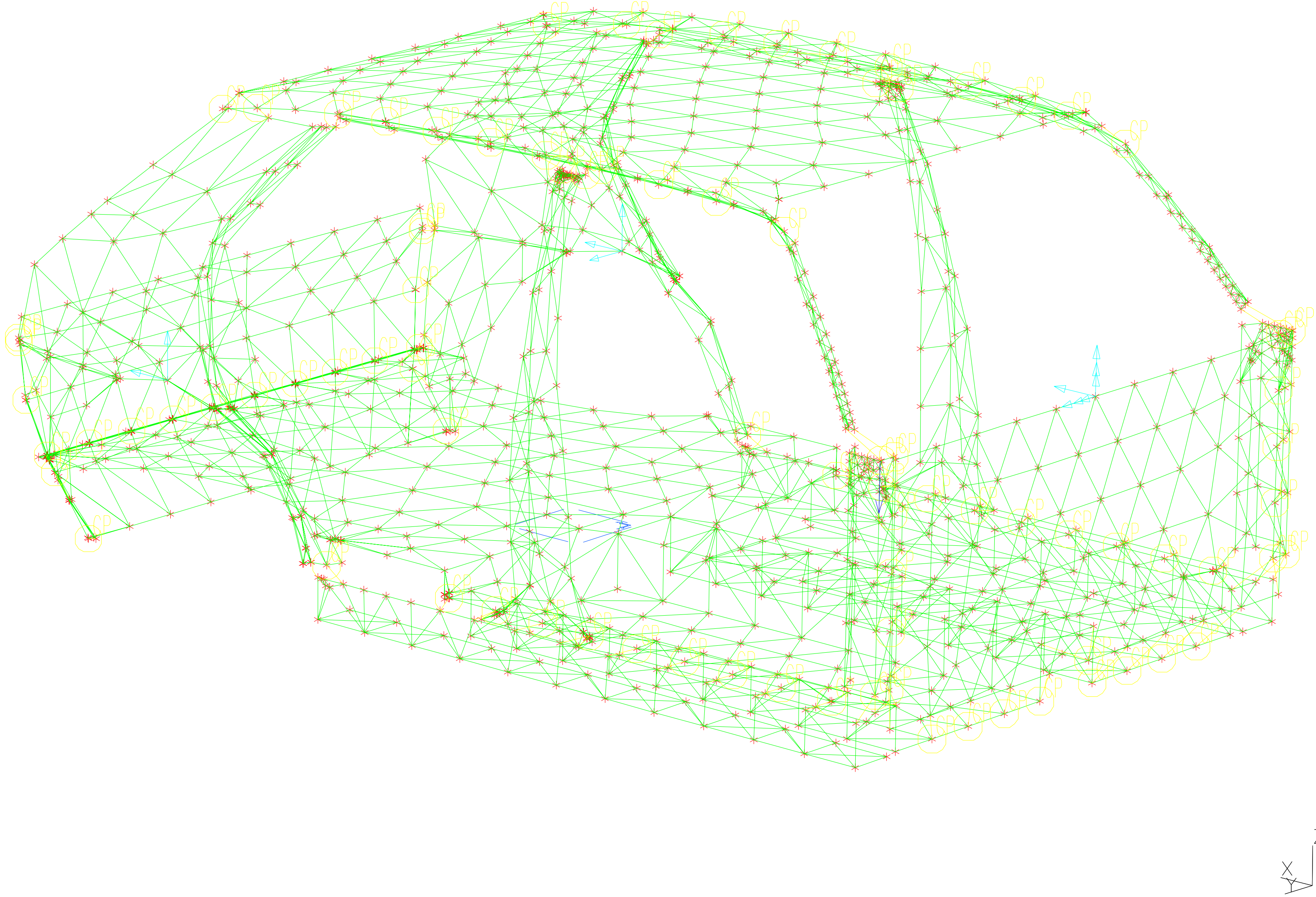
deflection (mm)	load (N)
0.00000	0.00
0.10160	284.67
0.17780	551.55
0.33020	836.22
0.78740	2001.60
0.91440	2286.27
1.16840	2837.82

y = mx + c
slope : 2424.16 N/mm
y - intercept : 51.79 N

y = mx + 0
slope : 2484.80 N/mm




Z
X


Z
X
Y

Problem: There's more to the problem!

- Information is *imprecise*
- Designer judgement and experience
- Some targets are *not explicit* (informal):
  - style
  - manufacturability
  - availability
- Uncontrolled variations (noise)
- Negotiations:
  - targets may change
  - how do targets interact?

Difficulties in Making the Decision Computable

Fuzziness: Specifications are imprecise.

Unmodelled Criteria: Many (often crucial) decisions depend on unquantified or unmeasured criteria.

Set-based design: Sets provide rapid design exploration; facilitate concurrency and support iteration.

Rationality: Need to reconcile competing, incommensurate objectives in a rational way.

Cost: Computation cost to effectively explore a large design space can be significant.

Difficulties in Making the Decision Computable

Fuzziness: Specifications are imprecise.

Unmodelled Criteria: Many (often crucial) decisions depend on unquantified or unmeasured criteria.

Set-based design: Sets provide rapid design exploration; facilitate concurrency and support iteration.

Rationality: Need to reconcile competing, incommensurate objectives in a rational way.

Cost: Computation cost to effectively explore a large design space can be significant.

Most of engineering, particularly design, can best be represented with some level of imprecision or approximation.

> *"Fuzziness is more than the exception in engineering design problems: usually there is no well-defined best solution or design."*

[Joseph A. Goguen, "L-Fuzzy Sets" *Journal Mathematical Analysis and Applications*, Volume 18, 1967, page 146.]

Imprecision is the representation an incomplete design description:
ranges of possibilities resulting from choices not yet made.

Preliminary design information is necessarily imprecise.

Designers need to evaluate designs early in the design process.

Designers need to rapidly explore large design spaces early in the design process.

Need to trade-off Precision *vs*. Computation Cost.

Functional Requirement $\mu_p(\vec{p})$:

the customer's expressed preference(s) on $\vec{p}$,

based on *performance considerations*.

Performance considerations are customer specifications and requirements.

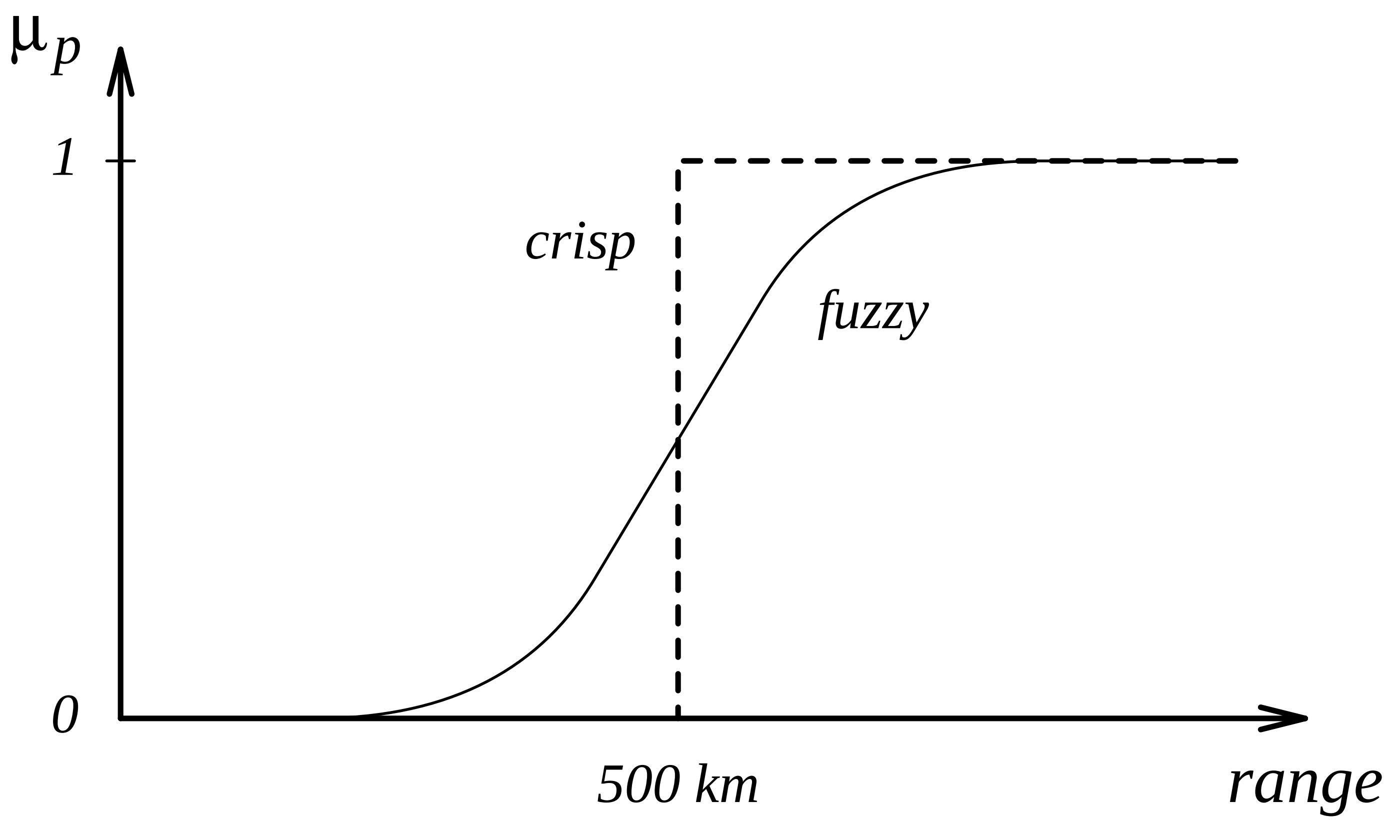
Design Preference $\mu_d(\vec{d})$:

the customer's un-expressed preference(s) on $\vec{d}$,

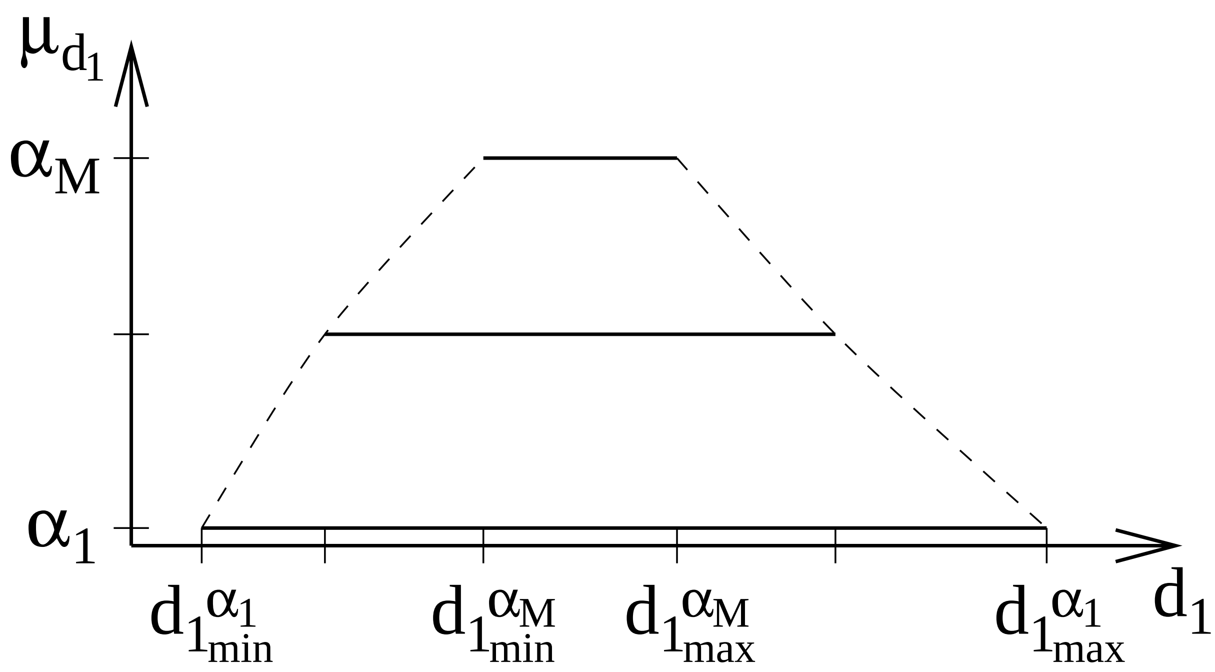
based on *design considerations*.

Design considerations are the aspects of performance not quantified by preferences on $\vec{p}$.

Fuzziness: Imprecise Specification

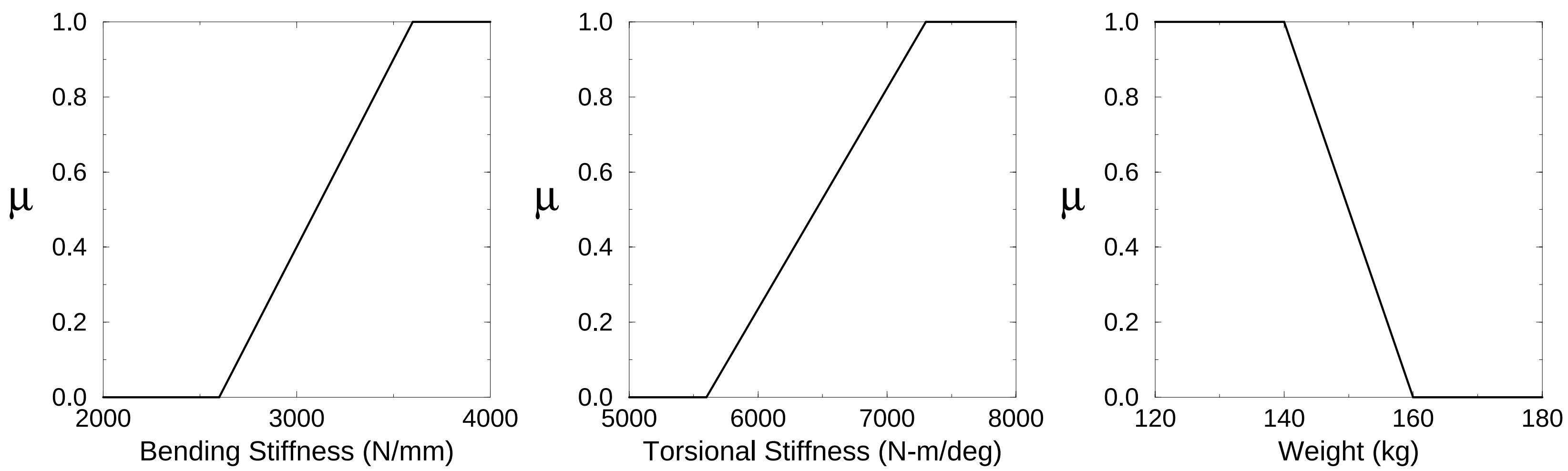


Fuzziness: Imprecise Design Variable



Calculating a comprehensive design problem

Use performance preferences in place of crisp targets:



Difficulties in Making the Decision Computable

Fuzziness: Specifications are imprecise.

Unmodelled Criteria: Many (often crucial) decisions depend on unquantified or unmeasured criteria.

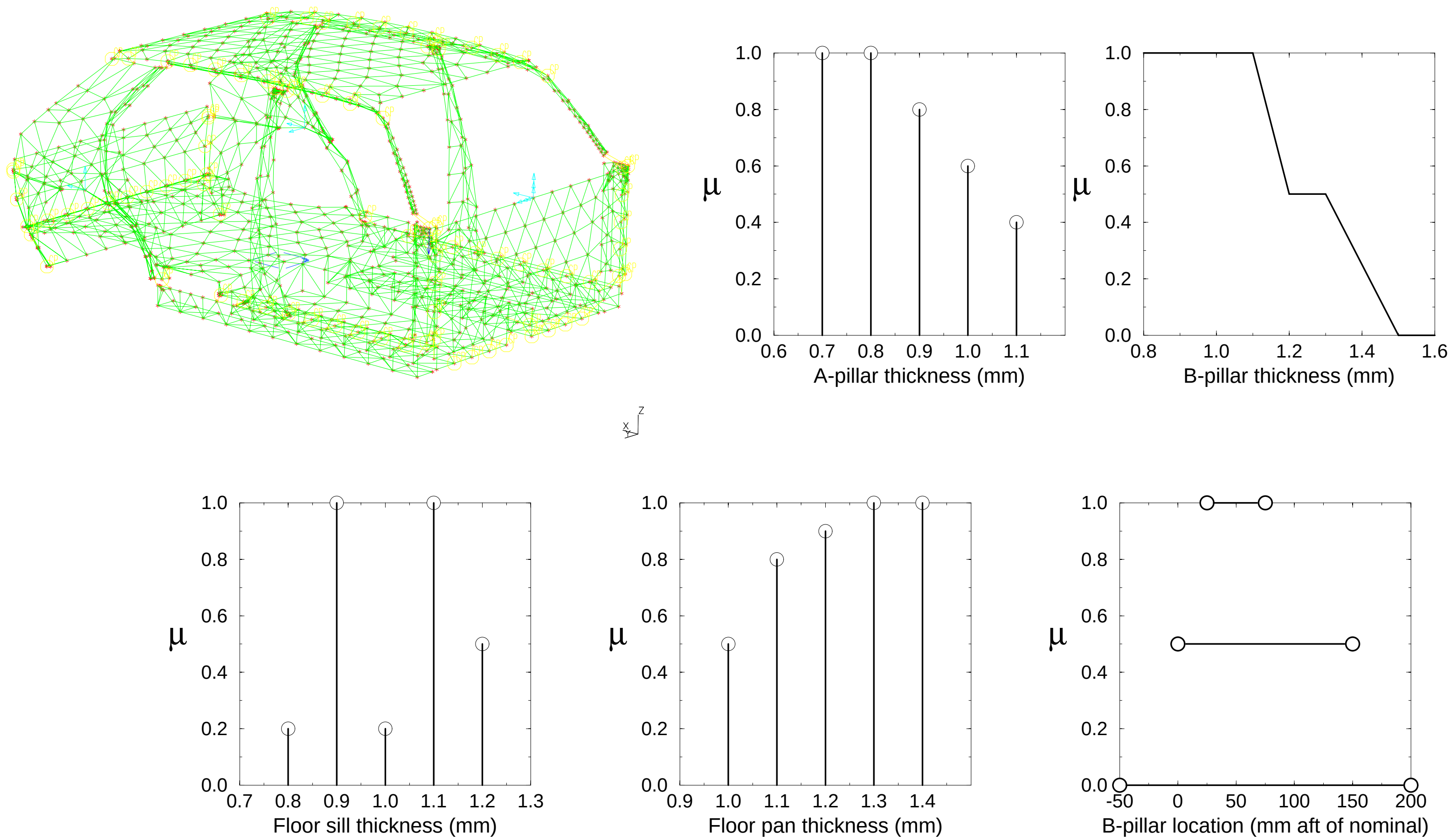
Set-based design: Sets provide rapid design exploration; facilitate concurrency and support iteration.

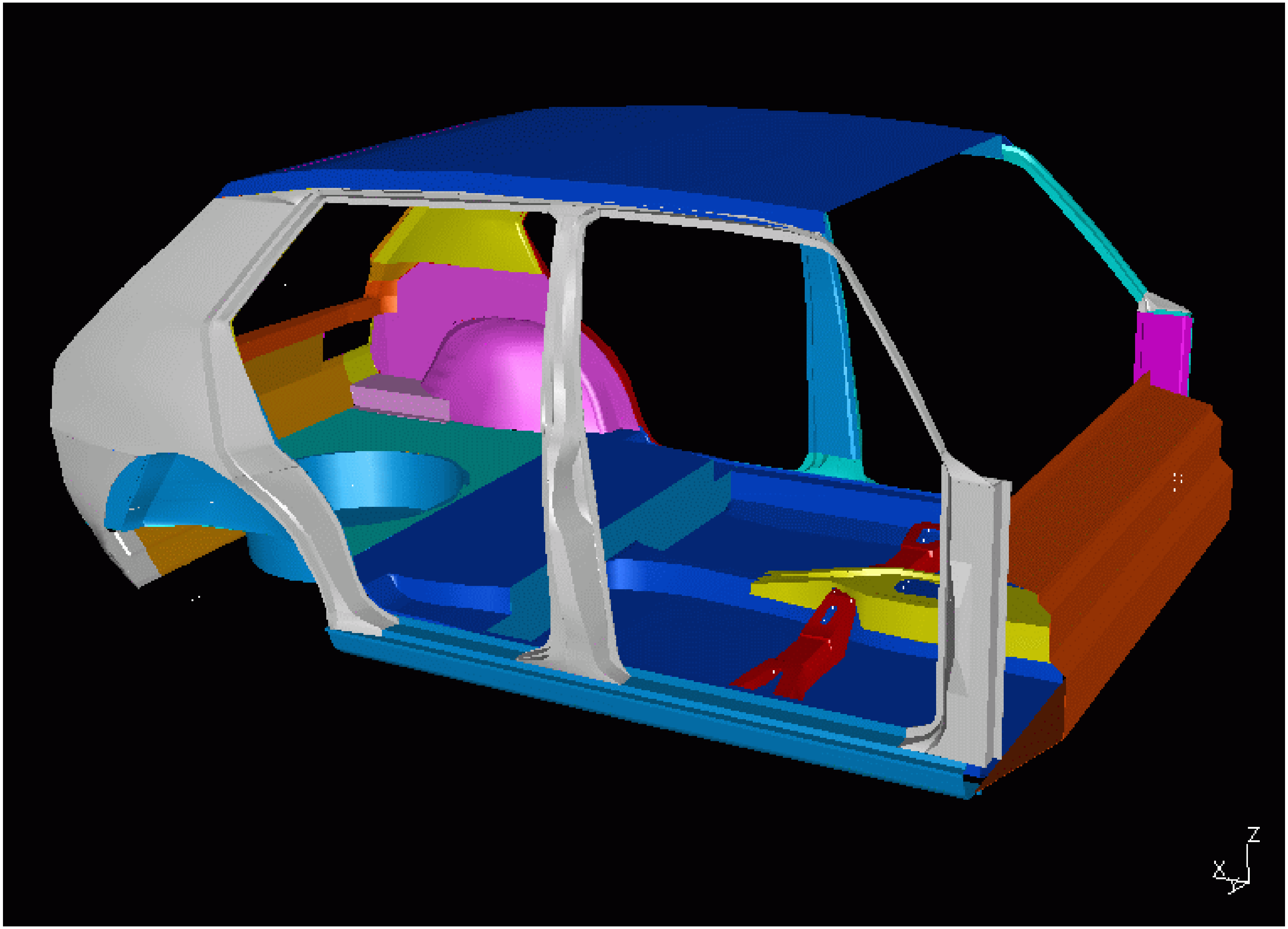
Rationality: Need to reconcile competing, incommensurate objectives in a rational way.

Cost: Computation cost to effectively explore a large design space can be significant.

Calculating a comprehensive design problem

Use designer preferences to incorporate unmodelled concerns




Z
X
Y

Difficulties in Making the Decision Computable

Fuzziness: Specifications are imprecise.

Unmodelled Criteria: Many (often crucial) decisions depend on unquantified or unmeasured criteria.

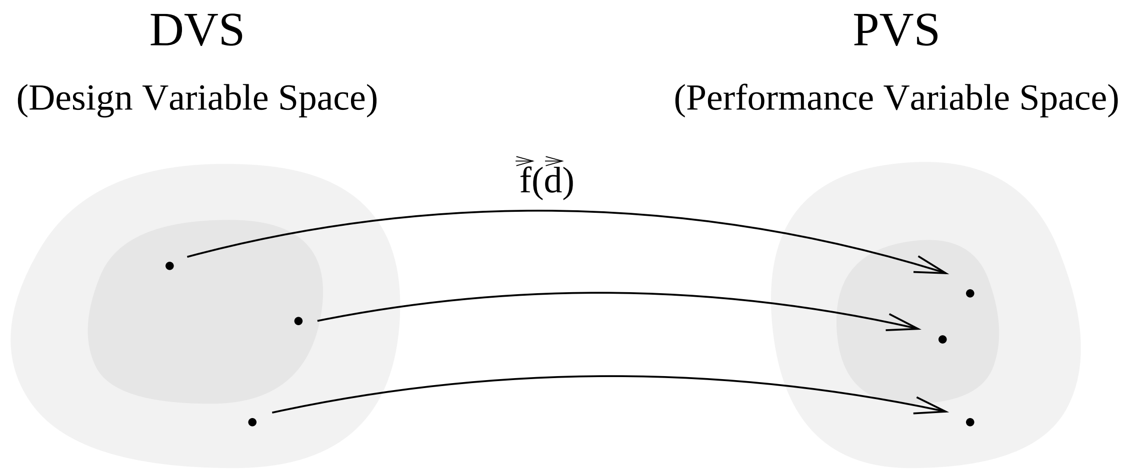
Set-based design: Sets provide rapid design exploration; facilitate concurrency and support iteration.

Rationality: Need to reconcile competing, incommensurate objectives in a rational way.

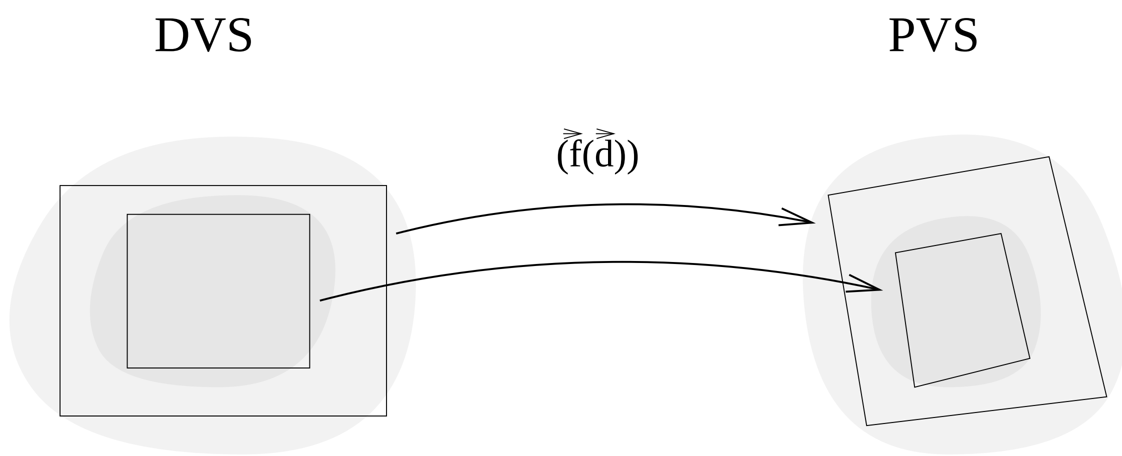
Cost: Computation cost to effectively explore a large design space can be significant.

Set-based Design

Traditional evaluation tools suppress imprecision,
and evaluate designs one at a time:



An alternative approach is to evaluate *sets* of designs:



Set-based Design

- Set refinement is a natural (& beneficial) way to develop designs.
  [Ward and Liker 1995]
- Intrinsically slower but has potential for time savings. [Eppinger 1991]
- Facilitates concurrency, communication, iterative redesign.
- There is a trade-off: not natural for optimization.

With this negotiation model:

- Level-set information is found with minimal extra computation.
- Sufficient conditions for propagation of set-based information.

The Basic Idea:

1. Designs are described by variables $\vec{d} \in$ *DVS*;
   Measured performances are given by $\vec{p} \in$ *PVS*.

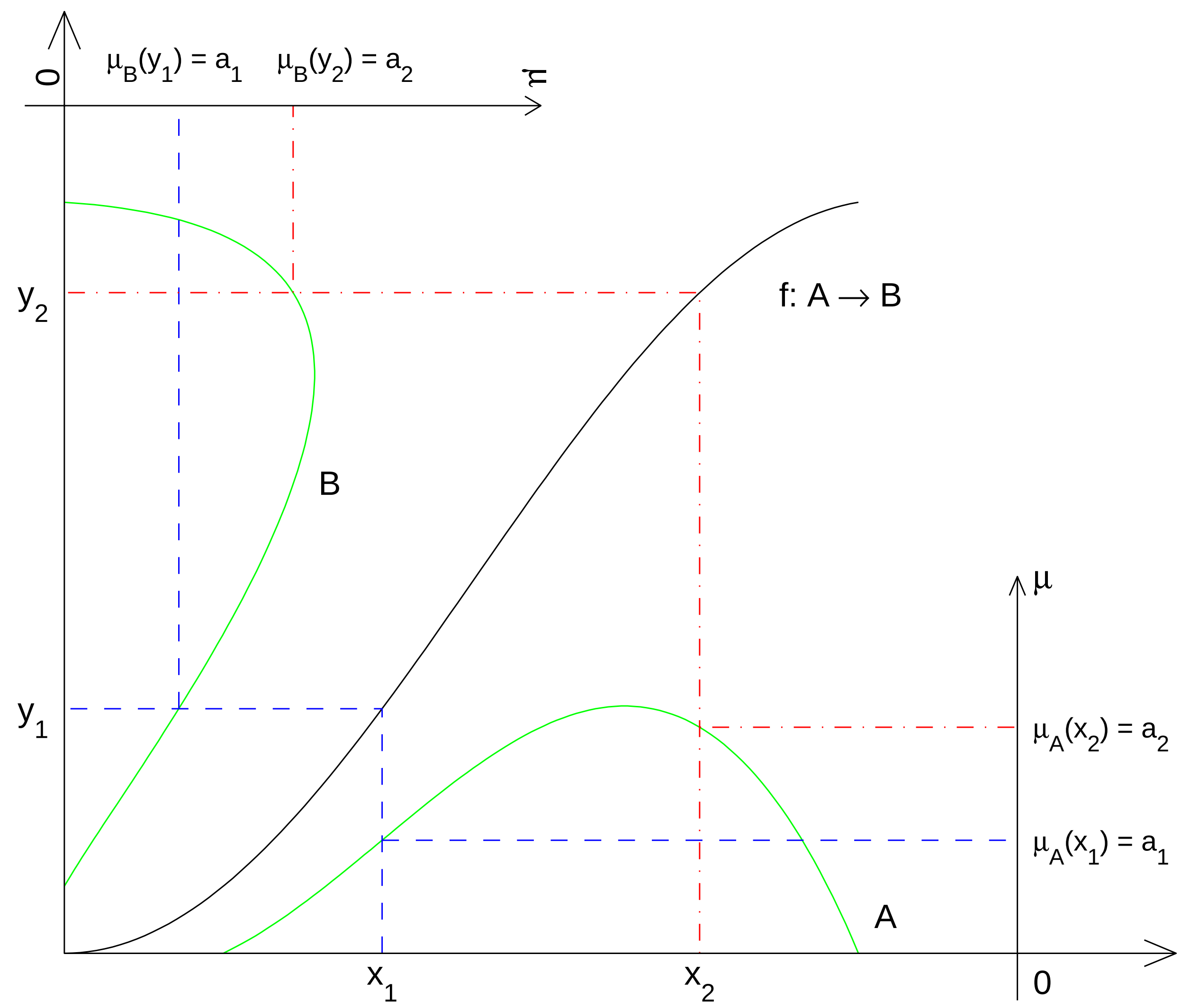
2. Specify preferences ($\mu \in [0, 1]$) on both *DVS* and *PVS*.

3. Map preferences from the design space (*DVS*) to the performance space (*PVS*) using the performance model f, and the extension principle [Zadeh 1975]:

$$\mu_d(p) = \sup_{\vec{d} \mid f(\vec{d})=p} \min_i \left(\mu_d(d_i)\right)$$

4. Aggregate results to determine overall preference.

[The Method of Imprecision (M$_o$I): Wood and Antonsson 1988, Otto 1992]



μ
0
$\mu_B(y_1) = a_1$
$\mu_B(y_2) = a_2$
y_2
y_1
f: A → B
B
A
μ
$\mu_A(x_2) = a_2$
$\mu_A(x_1) = a_1$
x_1
x_2
0

The *Method of Imprecision* ($\mathrm{M_oI}$)

Find the sets of designs $\vec{d}^*$ with the best performance:

$$f(\vec{d}^*) = \vec{p}^*$$

where $\vec{p}^*$ maximizes

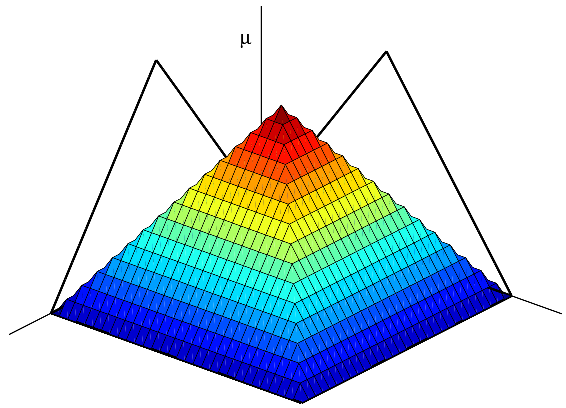
$$\mathcal{P}\left(\mu_d(\vec{d}), \mu_p(\vec{p})\right) =$$

$$\mathcal{P}\left(\mu_{d_1}(d_1), \ldots, \mu_{d_q}(d_q), \mu_{p_1}(p_1), \ldots, \mu_{p_n}(p_n)\right)$$

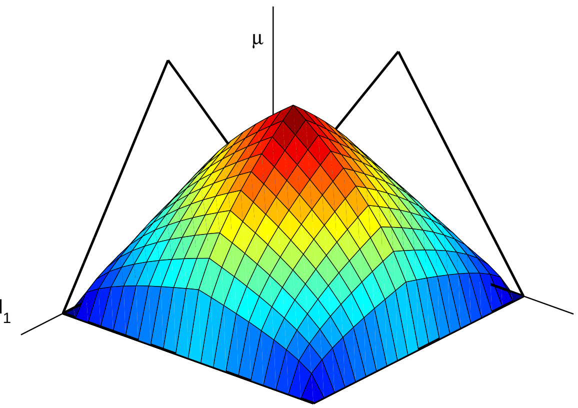
$\vec{d}$	q-vector of design variables
$\vec{p}$	n-vector of performance variables
$\mu_d(d_i)$	preference for ith design variable
$\mu_p(p_j)$	preference for jth measured aspect of performance
f	mapping from design to performance (analytic, black box, FEA, *etc.*)
$\mathcal{P}$	aggregation function — calculates overall preference

How should multiple preferences be aggregated?

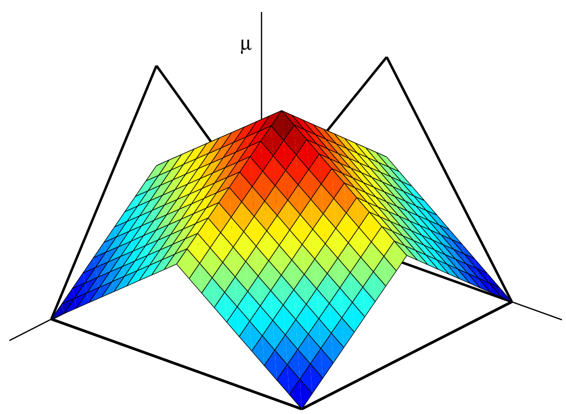
min



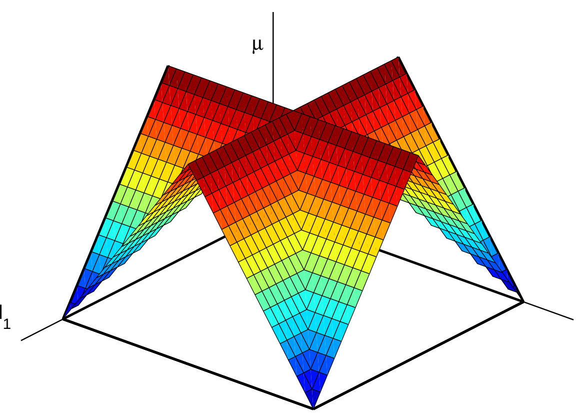
geometric mean



arithmetic mean



max



Difficulties in Making the Decision Computable

Fuzziness: Specifications are imprecise.

Unmodelled Criteria: Many (often crucial) decisions depend on unquantified or unmeasured criteria.

Set-based design: Sets provide rapid design exploration; facilitate concurrency and support iteration.

Rationality: Need to reconcile competing, incommensurate objectives in a rational way.

Cost: Computation cost to effectively explore a large design space can be significant.

Rationality and the Multi Attribute Decision Making problem (or, Why not Economics?):

- Designs are almost always judged on several criteria.
- These criteria compete (stiff *vs*. light) but do not compare (Newton/mm *vs*. kg?)
- Economics approach reduces everything to $$; introducing various difficulties:
  - Constraints
  - Non-probabilistic uncertainty
  - Practical issues in preference elicitation

[M. Tribus, *Rational Descriptions, Decisions, and Designs*, 1969]

[von Neumann and Morgenstern 1944, Tribus 1969, Keeney and Raiffa 1976, Bradley and Agogino 1991, Thurston 1991.]

Aggregation Operator Axioms

At each point $\vec{x}$ the following hold:

1. **Monotonicity:**
   $\mathcal{P}(\mu_1, \mu_2; \omega_1, \omega_2)(\vec{x}) \leq \mathcal{P}(\mu_1, \mu_2'; \omega_1, \omega_2)(\vec{x}) \quad \forall\, \mu_2(\vec{x}) \leq \mu_2'(\vec{x})$
   $\mathcal{P}(\mu_1, \mu_2; \omega_1, \omega_2)(\vec{x}) \leq \mathcal{P}(\mu_1, \mu_2; \omega_1, \omega_2')(\vec{x}) \quad \forall\, \omega_2 \leq \omega_2'; \;\; \mu_1(\vec{x}) < \mu_2(\vec{x})$

2. **Symmetry:**
   $\mathcal{P}(\mu_1, \mu_2; \omega_1, \omega_2)(\vec{x}) = \mathcal{P}(\mu_2, \mu_1; \omega_2, \omega_1)(\vec{x})$

3. **Continuity:**
   $\mathcal{P}(\mu_1, \mu_2; \omega_1, \omega_2)(\vec{x}) = \lim_{\mu_2'(\vec{x}) \to \mu_2(\vec{x})} \mathcal{P}(\mu_1, \mu_2'; \omega_1, \omega_2)(\vec{x})$
   $\mathcal{P}(\mu_1, \mu_2; \omega_1, \omega_2)(\vec{x}) = \lim_{\omega_2' \to \omega_2} \mathcal{P}(\mu_1, \mu_2; \omega_1, \omega_2')(\vec{x})$

4. **Idempotency:**
   $\mathcal{P}(\mu, \mu; \omega_1, \omega_2)(\vec{x}) = \mu(\vec{x}) \;\; \forall\, \omega_1 + \omega_2 > 0$

5. **Annihilation:**
   $\mathcal{P}(\mu, 0; \omega_1, \omega_2)(\vec{x}) = 0 \;\; \forall\, \omega_2 \neq 0$

6. **Self-scaling weights:**
   $\mathcal{P}(\mu_1, \mu_2; \omega_1 t, \omega_2 t)(\vec{x}) = \mathcal{P}(\mu_1, \mu_2; \omega_1, \omega_2)(\vec{x}) \quad \forall\, \omega_1 + \omega_2, t > 0$

7. **Zero weights:**
   $\mathcal{P}(\mu_1, \mu_2; \omega_1, 0)(\vec{x}) = \mu_1(\vec{x}) \;\; \forall\, \omega_1 \neq 0$

Trade-off Strategies and Weights

Trade-off strategies reflect the compensation among goals inherent in the problem (independent of weights).

- A *compensating* strategy is used when higher preference in one variable may compensate for lower preference in another.

- A *non-compensating* strategy is appropriate when the overall performance of a design is limited by its lowest-performing attribute.

Weights reflect relative importance of attributes.

A strategy–weight pair is required to define the aggregation of two independent attributes.

Early M$_o$I Aggregation Functions

[Otto and Antonsson 1990]

non-compensating: $\mathcal{P}(\mu_1, \cdots, \mu_n; \omega_1, \ldots, \omega_n) = \min(\mu_1, \cdots, \mu_n)$

and compensating:

$$\mathcal{P}(\mu_1, \cdots, \mu_n; \omega_1, \ldots, \omega_n) = ({\mu_1}^{\omega_1} \cdots {\mu_n}^{\omega_n})^{\frac{1}{\omega_1 + \cdots + \omega_n}}$$

Other functions (QFD [Hauser and Clausing 1988], AHP [Saaty 1980], fuzzy logic):

- weighted sums:

$$u(x^1, \ldots, x^N) = \sum_{i=1}^{N} \omega_i \frac{u(x^i)}{N}.$$

- T-norms ($<$ *min*) model intersection (logical AND).
- T-conorms ($>$ *max*) model union (logical OR).

Weighted Means are a class of functions

$$\mathcal{P}(\mu_1, \mu_2; \omega_1, \omega_2)$$

that satisfy all the M$_\text{O}$I axioms except annihilation and also have strict monotonicity.

Theorem The properties of the weighted mean are necessary and sufficient for the function $\mathcal{P}(\mu_1, \mu_2; \omega_1, \omega_2)$ to be of the form

$$\mathcal{P}(\mu_1, \mu_2; \omega_1, \omega_2) = g\left(\frac{\omega_1 g^{-1}(\mu_1) + \omega_2 g^{-1}(\mu_2)}{\omega_1 + \omega_2}\right)$$

where $\exists\ \mu_a, \mu_b$ such that

$$\mu_a \leq \mu_1, \mu_2 \leq \mu_b\ ;\ \omega_1, \omega_2 \geq 0\ ; \omega_1 + \omega_2 > 0$$

and g is a strictly monotonic, continuous function with inverse g^{-1}.

Proof see [J. Aczél, *Lectures on Functional Equations*, 1966].

A parameterized family of aggregation functions

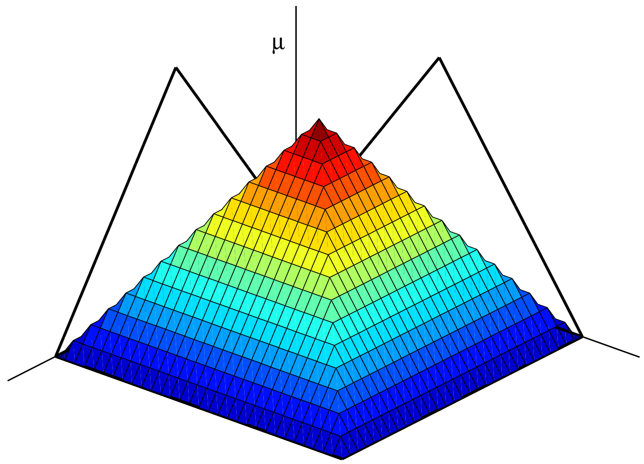
$g(t) = t^s$, where s is a real parameter, generates the aggregation function:

$$\mathcal{P}_s(\mu_1, \mu_2; \omega_1, \omega_2) = \left(\frac{\omega_1 {\mu_1}^s + \omega_2 {\mu_2}^s}{\omega_1 + \omega_2} \right)^{\frac{1}{s}}$$

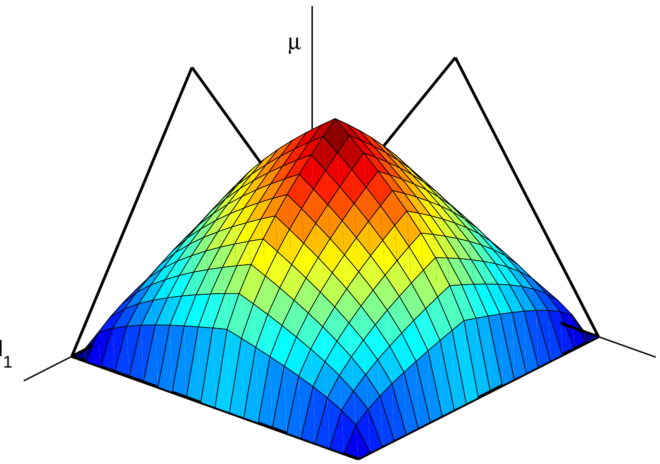
A little algebra shows the following:

$$\begin{array}{lcccl}
\mathcal{P}_0 & = & \lim_{s \to 0} \mathcal{P}_s & = & \text{geometric mean} \\
\mathcal{P}_{-\infty} & = & \lim_{s \to -\infty} \mathcal{P}_s & = & \textit{min} \\
\mathcal{P}_{\infty} & = & \lim_{s \to +\infty} \mathcal{P}_s & = & \textit{max} \\
\mathcal{P}_1 & = & \frac{\omega_1 \mu_1 + \omega_2 \mu_2}{\omega_1 + \omega_2} & = & \text{arithmetic mean}
\end{array}$$

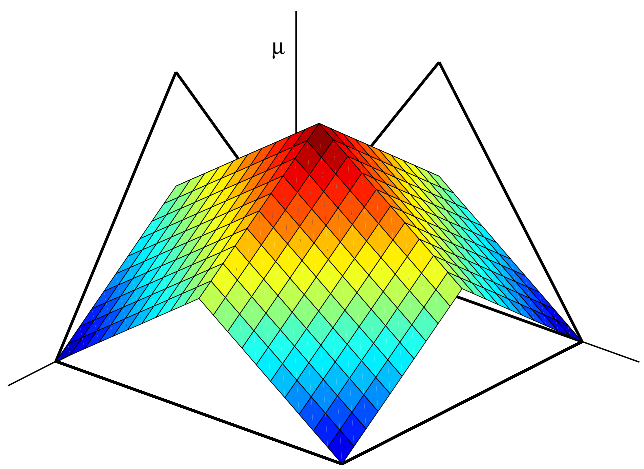
$\mathcal{P}_{-\infty}$ (min)



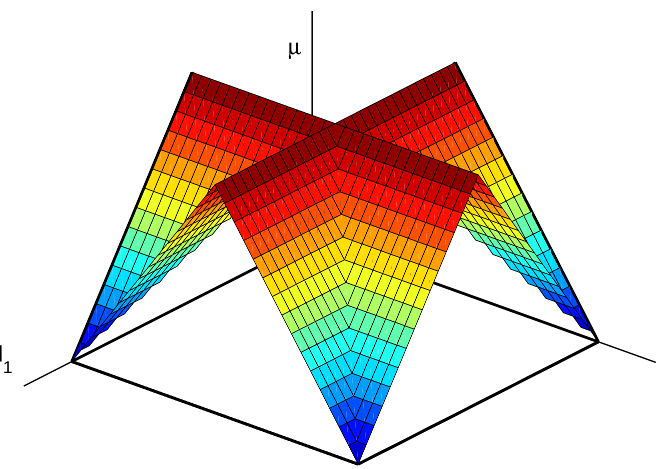
$\mathcal{P}_0$ (geometric mean)



$\mathcal{P}_1$ (arithmetic mean)



$\mathcal{P}_{\infty}$ (max)



$\mathcal{P}_s$ Example

[H.-J. Zimmermann, *Fuzzy Set Theory,* 1985]

A company makes two products, 1 and 2:

- Product 1 yields \$2 profit but requires \$1 in imports.
- Product 2 can be exported for \$2 revenue but makes only \$1 profit.

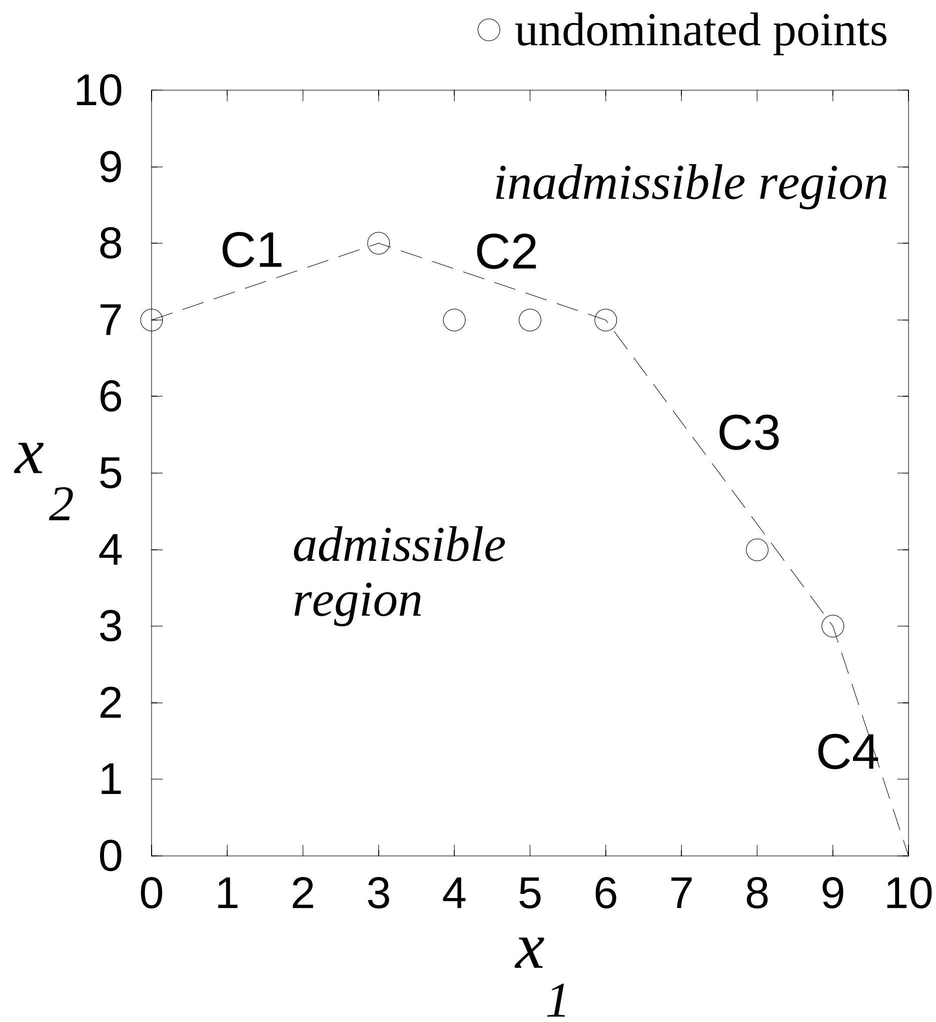
$x_1 =$ Number of Product 1 produced.
$x_2 =$ Number of Product 2 produced.

Objective: "Maximize" balance of trade (z_1) and profits (z_2):

$$\vec{z}(\vec{x}) = \begin{pmatrix} -1 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

subject to a number of constraints:

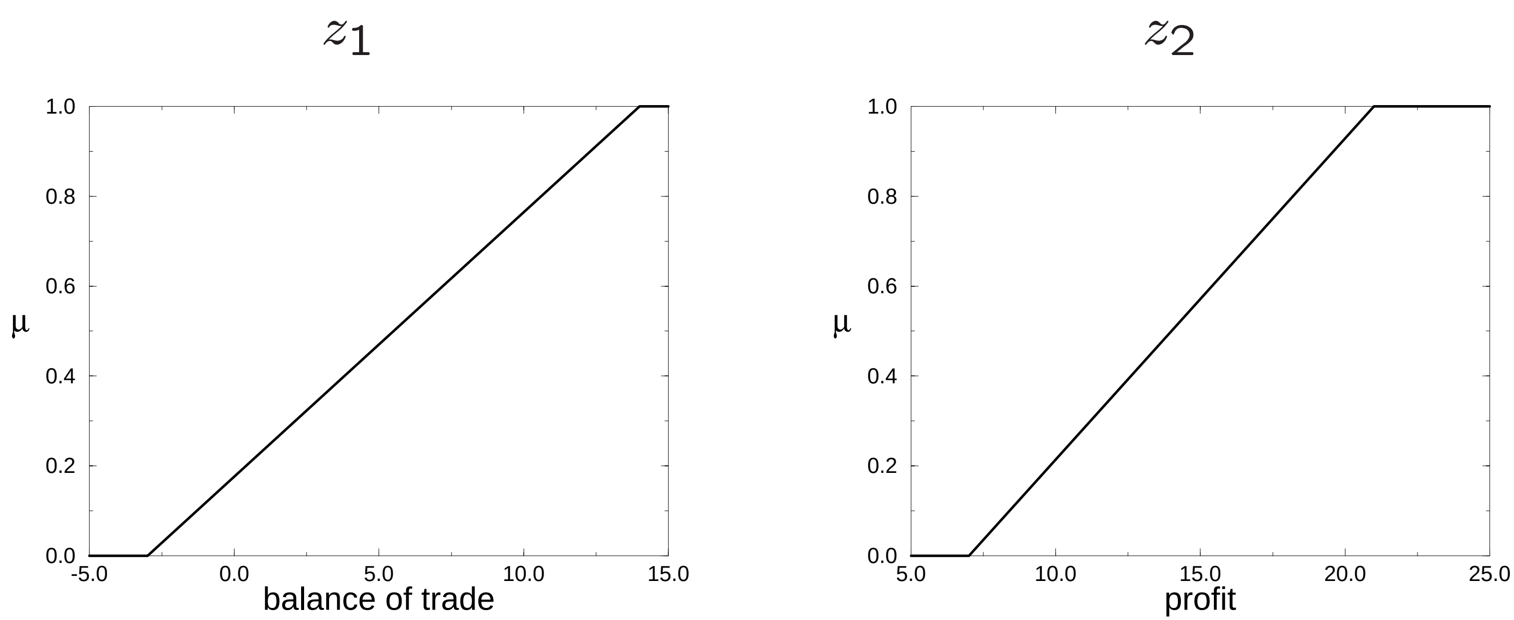
Decision Space with Pareto-optimal Region:



Constraints:

C1:	$-x_1 + 3x_2$	$\leq$	21
C2:	$x_1 + 3x_2$	$\leq$	27
C3:	$4x_1 + 3x_2$	$\leq$	45
C4:	$3x_1 + x_2$	$\leq$	30
C5:	x_1	$\geq$	0
C6:	x_2	$\geq$	0

Performance preferences:



Note: $\vec{z}\begin{pmatrix} 0 \\ 7 \end{pmatrix} = \begin{pmatrix} 14 \\ 7 \end{pmatrix}, \quad \vec{z}\begin{pmatrix} 9 \\ 3 \end{pmatrix} = \begin{pmatrix} -3 \\ 21 \end{pmatrix}$

Some solutions:

Zimmermann uses the *min* ($\mathcal{P}_{-\infty}$):

$\vec{x} = (5.03, 7.32)$ is best to two decimal places

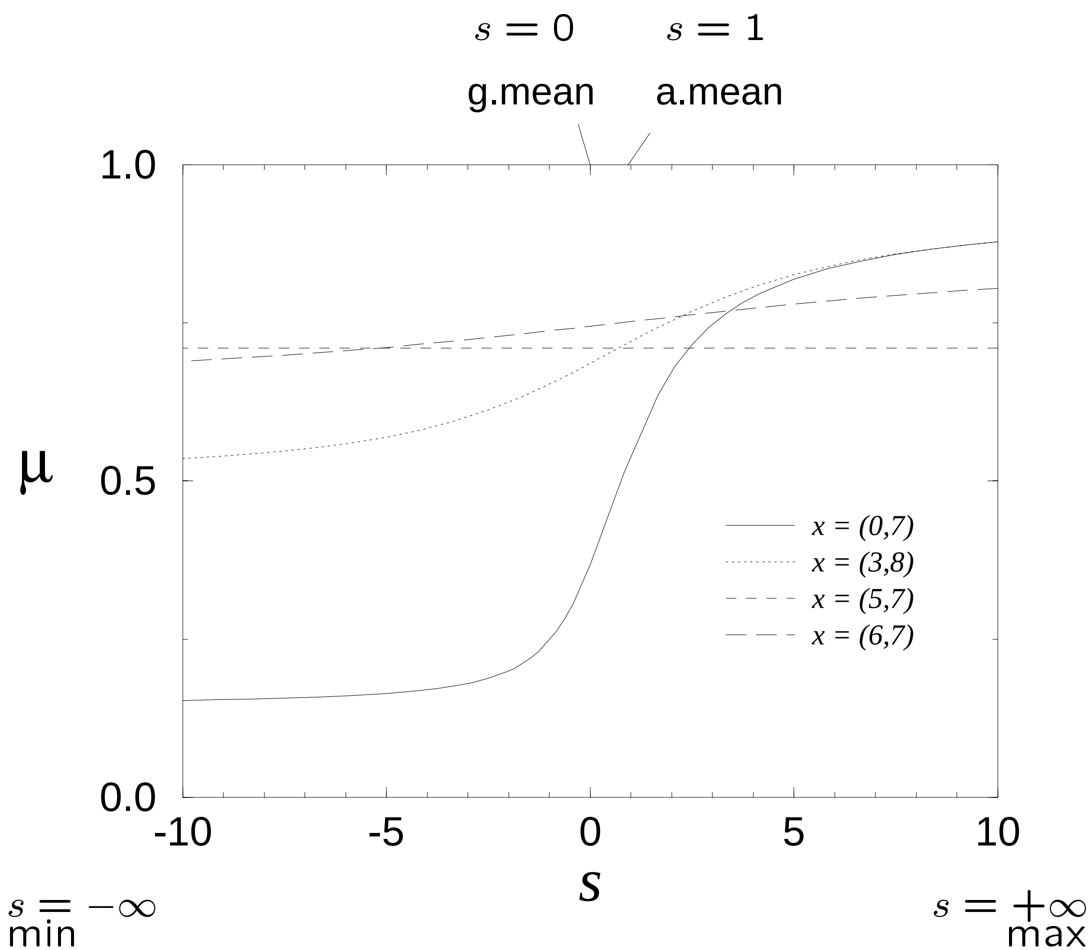
$\vec{x} = (5, 7)$ is the best integer value

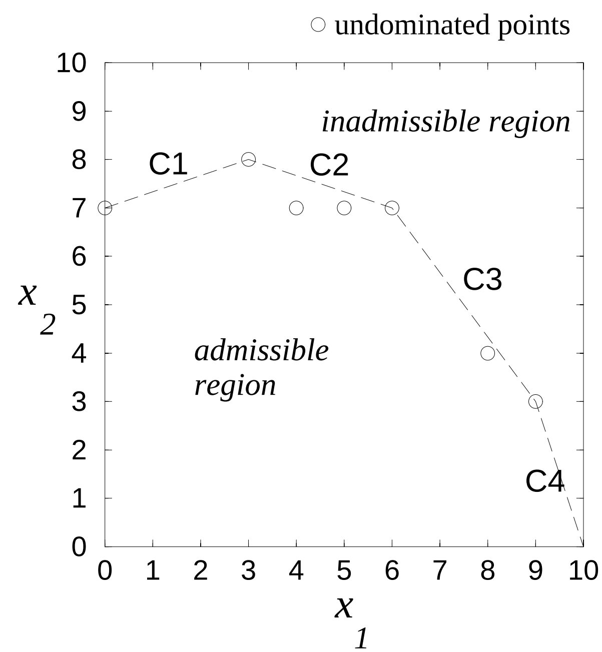
For the geometric mean ($\mathcal{P}_0$):

$\vec{x} = (5.70, 7.10)$ is best to two decimal places

$\vec{x} = (6, 7)$ is the best integer value .

Optimal points for different aggregation functions $\mathcal{P}_s$





The decision depends on the aggregation.

Use indifference points to determine strategies and weights.

Definition:

A **Negotiation** (or a trade-off) is the choice of the compensation parameter (or strategy) s to aggregate multiple preferences, and the weights ω.

For any undominated (Pareto-optimal) design, there is a choice of s and ω that selects that point over all other undominated points.

This is not true for methods that use fixed strategies (weighted sum, for example).

Choosing a strategy and a pair of weights

State of the art:

- specify weights directly, with fixed granularity (AHP, QFD)
- extract eigenvectors from matrix, normalize (AHP)
- no choice of strategies (usually arithmetic mean)

This negotiation model calculates a strategy and a weight pair using indifference points. For two preferences:

- Get $\mu = 0.5$, $\mu = 1.0$ for each attribute.
- We know $\mathcal{P}(\mu_1, \mu_2) = 0.5$ for $\mu_1 = \mu_2 = 0.5$.
- Ask, “At what value of μ_1 is there indifference between $\mathcal{P}(\mu_1, 1.0)$ and $\mathcal{P}(0.5, 0.5) = 0.5$?”
- Simple procedure (with some numerical solving) returns s and $\frac{\omega_1}{\omega_2}$.

1. Determine x and y such that $\mathcal{P}_s(x,1) = \mathcal{P}_s(1,y) = 0.5$.
2. Let $b = \frac{\omega_y}{\omega_x}$.
3. If $x = y$, then $b = 1$:
   (a) If $x = 0.5$, then $s = -\infty$.
   (b) If $x = 0.25$, then $s = 0$.
   (c) If $x > 0.25$, then $s \in (-\infty, 0)$. Solve $x^s + 1 = 2(0.5)^s$ numerically.
   (d) If $x < 0.25$, then $s \in (0, \infty)$. Solve $x^s + 1 = 2(0.5)^s$ numerically.
4. If $x \neq y$, then $b \neq 1$. Note that if $s = 0$,

   $$x^m = 0.5 = y^{1-m} \Rightarrow y^{1-\log_x 0.5} = 0.5$$

   (a) If $y^{1-\log_x 0.5} = 0.5$, then $s = 0$, and $b = \frac{1-\log_x 0.5}{\log_x 0.5}$
   (b) If $y^{1-\log_x 0.5} > 0.5$, then $s < 0$.
   If $y^{1-\log_x 0.5} < 0.5$, then $s > 0$.
   Solve numerically for s from
   $\left(\frac{1+by^s}{1+b}\right)^{\frac{1}{s}} = \left(\frac{x^s+b}{1+b}\right)^{\frac{1}{s}} = 0.5$
   which reduces to
   $(x^s - 0.5^s)(y^s - 0.5^s) = (1 - 0.5^s)^2$
   From s, determine b.

It should be noted that this method will never return an answer $s > 1$, which is nice since it avoids “supercompensating” functions. It should also be noted that if either x or y is close to 0, the (s, b) pair is quite sensitive to small differences in x and y. In these cases, it might be preferable to elicit other indifference points to determine s and b.

Saaty’s AHP could be used in pairwise comparison at the start, as a check. There is (presently) no comparable normalization scheme for strategies s.

Choosing a strategy and a pair of weights, Example

Bending stiffness *vs*. torsional stiffness:

$\mu_1(3600) = \mu_2(7200) = 1$

$\mu_1(3000) = \mu_2(6400) = 0.5$

$\mathcal{P}_s(0.5, 1) \approx \mathcal{P}_s(1, 0.5)$

$\quad \approx \mathcal{P}_s(0.5, 0.5) = 0.5$

Conclude:

$s = -\infty, \omega_2 = \omega_1$

B-pillar location (style) *vs*. other design preferences:

$\mathcal{P}_s(0.4, 1) = \mathcal{P}_s(1, 0.3) = 0.5$

Conclude:

$s = -1.4, \omega_2 = 0.6\omega_1$

Quantified *vs*. Unquantified:

$\mathcal{P}_s(0.3, 1) = \mathcal{P}_s(1, 0.2) = 0.5$

Conclude:

$s = -0.02, \omega_2 = 0.8\omega_1$

Uncontrolled Variations (Noise)

- For typical noise, the preference of a design parameter set is weighted by its probability of occurring through the probabilistic uncertainty.
- Given a probability space (NPS), the expected preference of a point $d \in \text{DPS}$ is defined by:

$$E[\mu(d)] = \int_{\text{NPS}} \mu(d) \; \delta Pr$$

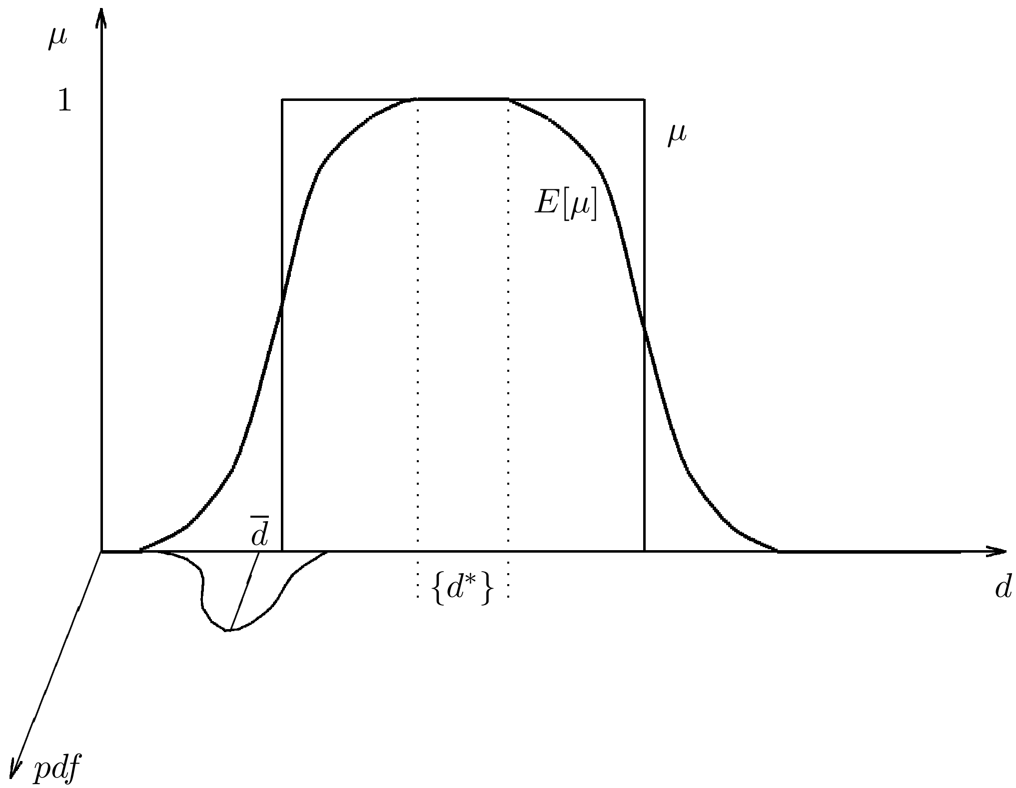
Uncontrolled Variations (Noise)

- If the NPS $\simeq \mathbb{R}$, then a probability density function $pdf(n)$ would be used.
- The integral of the equation above then becomes:

$$E[\mu(d)] = \int_{\text{NPS}|d} \mu(d)\; pdf(n|d)\; \delta(n|d)$$

- Since the distributions over n can vary with d, the notation $n|d$ is used.

Uncontrolled Variations (Noise)



Difficulties in Making the Decision Computable

Fuzziness: Specifications are imprecise.

Unmodelled Criteria: Many (often crucial) decisions depend on unquantified or unmeasured criteria.

Set-based design: Sets provide rapid design exploration; facilitate concurrency and support iteration.

Rationality: Need to reconcile competing, incommensurate objectives in a rational way.

Cost: Computation cost to effectively explore a large design space can be significant.

Computation Cost Issues:

- Real problems have “many” (10+) design variables.
- Real problems can have expensive analysis functions.

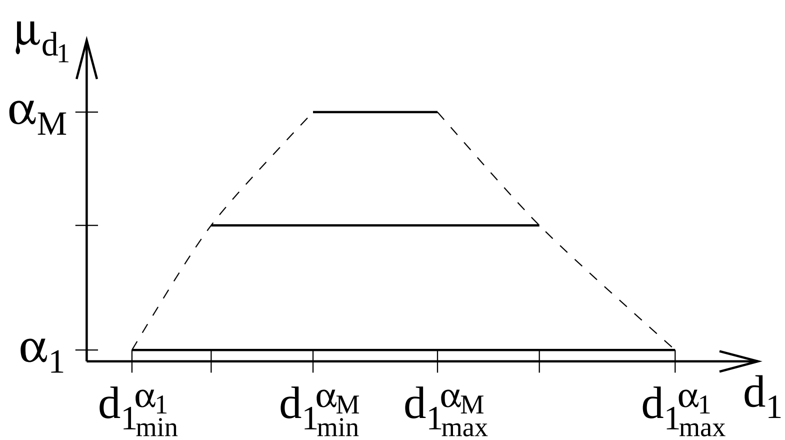
Some solutions:

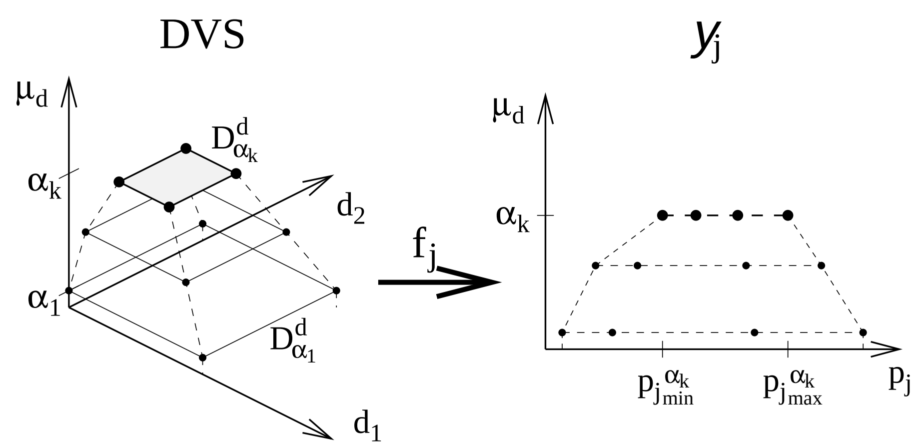
- Apply approximation methods to analysis function f [Law 1996]
- Technical (not substantial) restrictions on μ
- α-cut representations

Set-Based Computation Cost

Wrapper around existing analysis methods (f).

The *Level Interval Algorithm* (*LIA*) discretizes design preference into α-cut intervals:





Construct a linear approximation to $\vec{f}$ over D_ϵ^d where possible:

[Law and Antonsson 1995, 1996]

$$\vec{f'}(\vec{d}) = \vec{f}(\vec{d}_{\text{ctr}}) + \vec{\Delta} + \mathbf{A}[\vec{d} - \vec{d}_{\text{ctr}}]$$

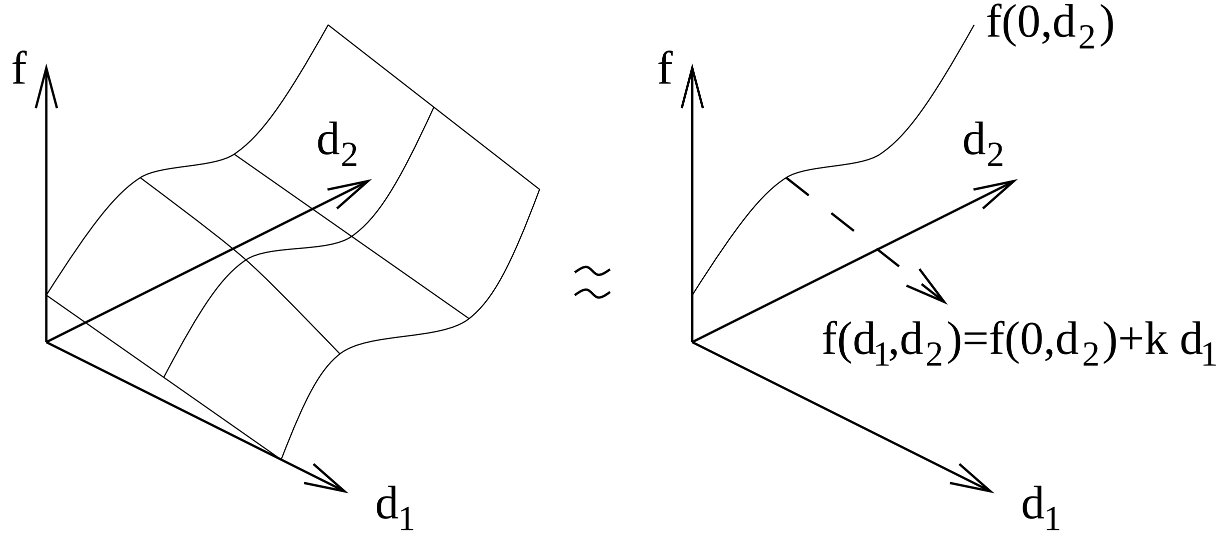
Where a linear approximation fails (*e.g.*, non-monotonicity),

find extremal points (*e.g.*, using Thompson's method),

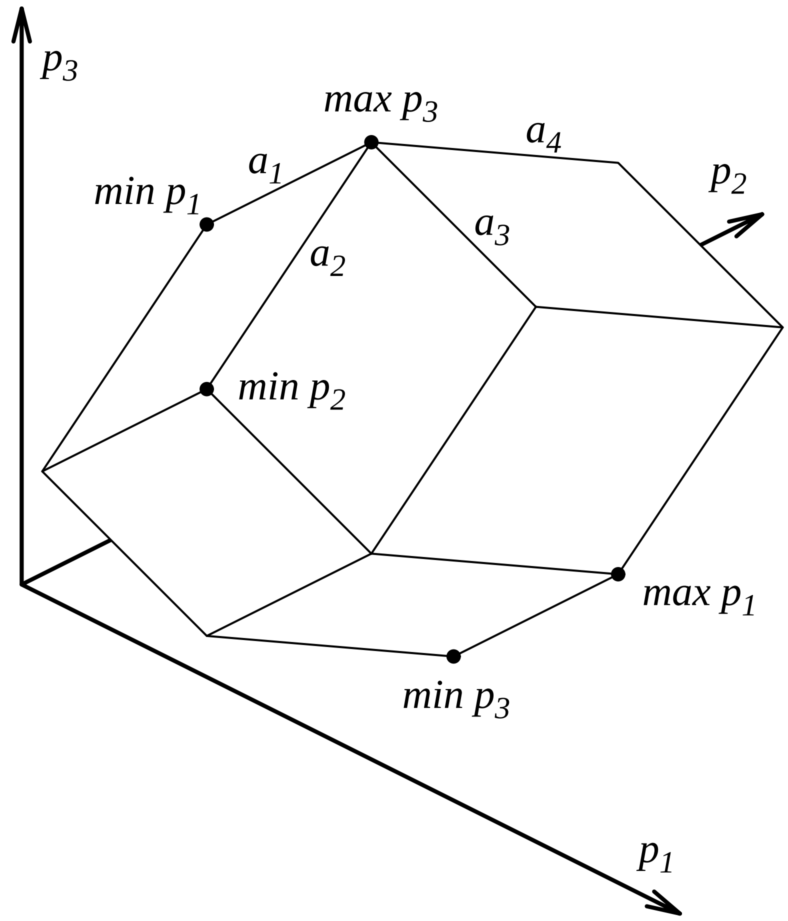
then build a linear approximation between extremal points.

A linear approximation $\vec{f}'$ fulfills several purposes:

1. It removes near-linear design variables from the search space for optimization.



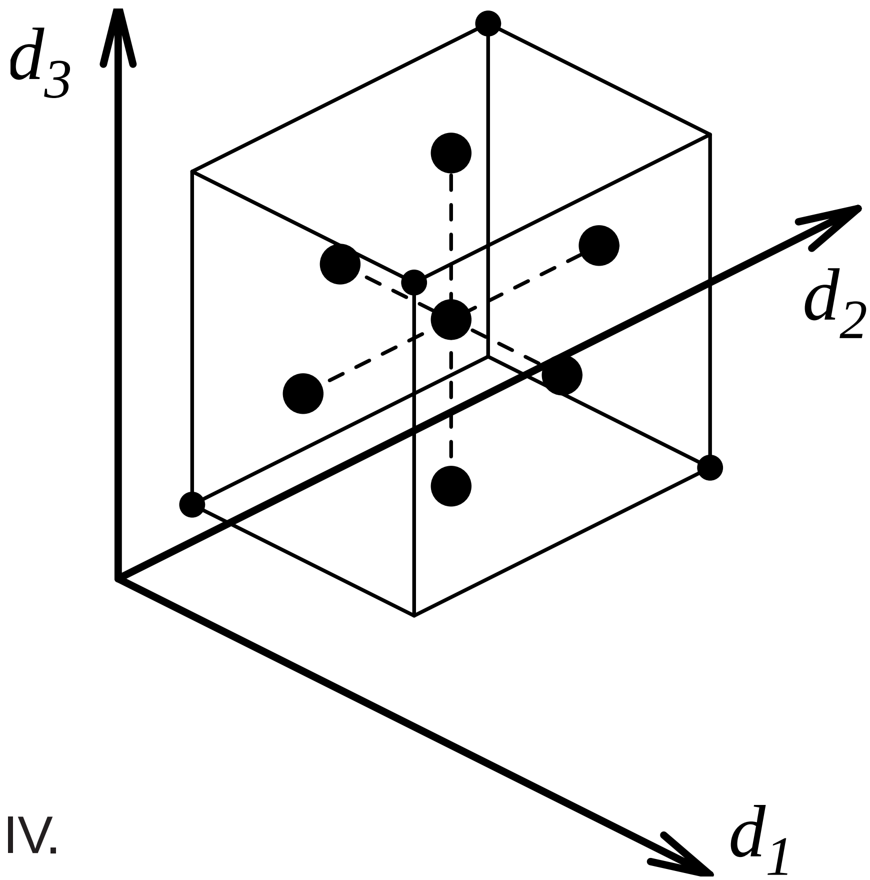
2. It allows the geometry of $P^d_{\alpha_k}$ to be interpolated between extremal points.



3. It provides a means to backwards map $\mu_p(\vec{p})$ onto the *DVS*.

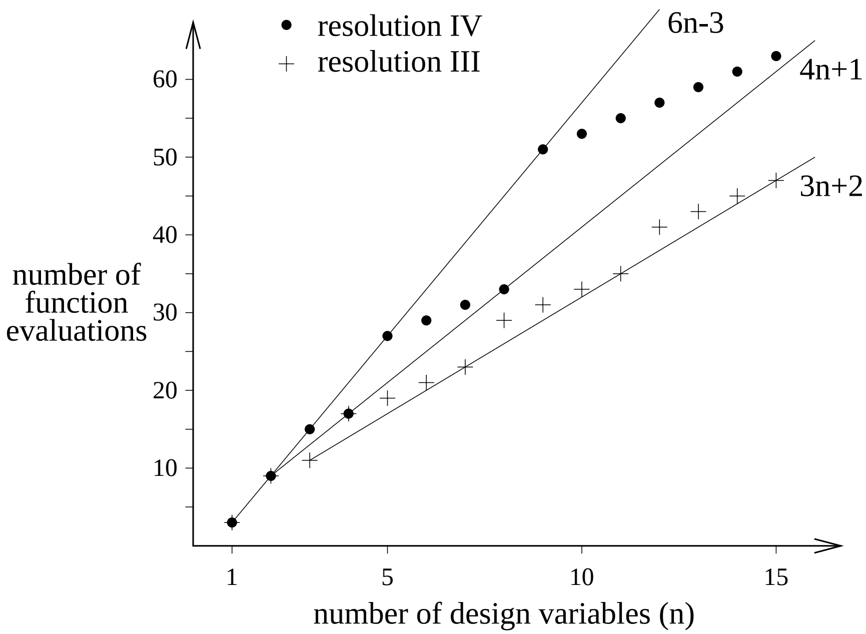
Design of Experiments (DOE) is used to construct $\vec{f}'$.

Central Composite Design:



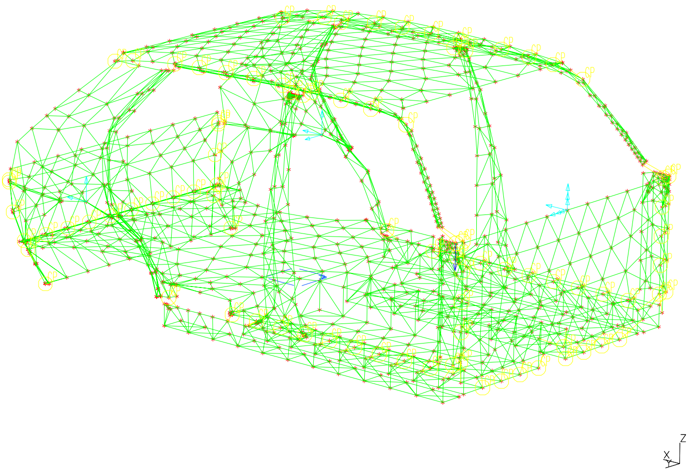
Resolution: III *vs*. IV.

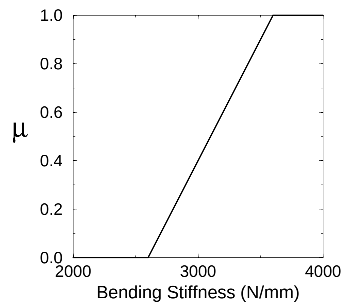
Computation Cost:

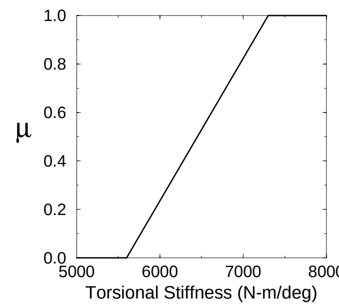


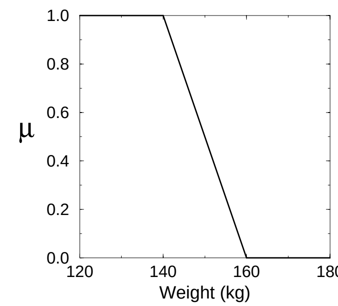
For a Central Composite Design.

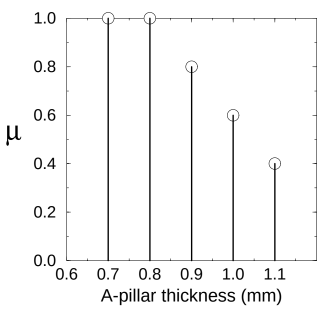
Return to the VW Example

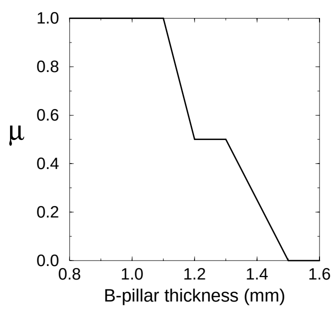


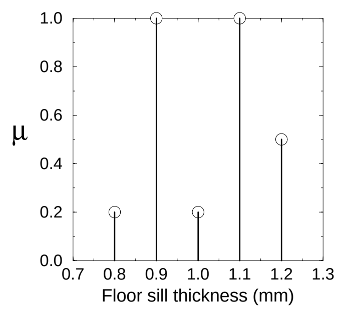


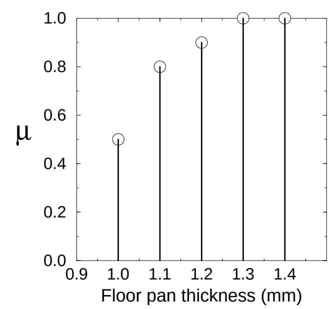


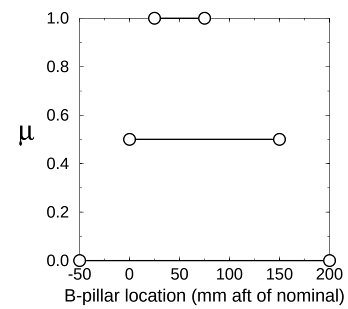




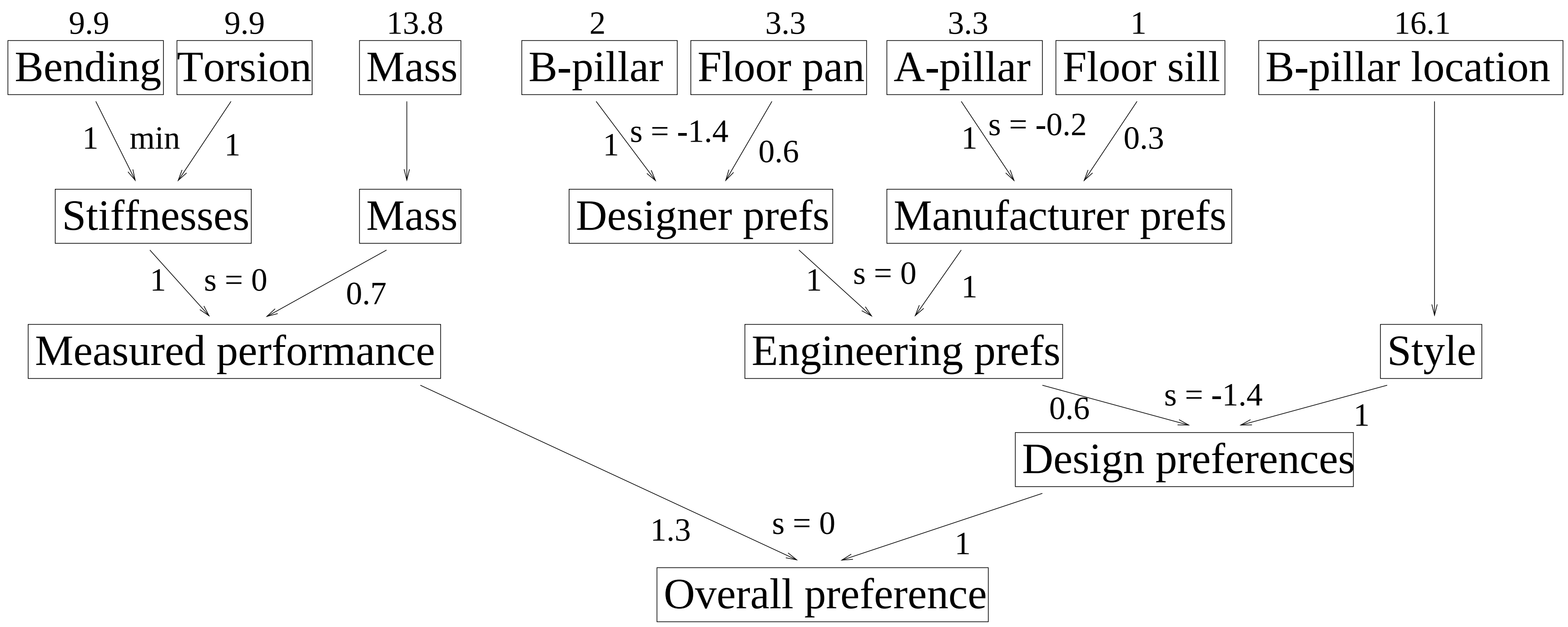








Hierarchical aggregation



The new maximization problem:

$$\mu_o(\vec{d}) = \mathcal{P}_0\left(\mu_d, \mu_p; 1, 1.3\right)$$

where

$$\mu_d =$$

$$\mathcal{P}_{-1.4}\left(\mathcal{P}_0\left(\mathcal{P}_{-1.4}\left(d_2, d_4; 1, 0.6\right), \mathcal{P}_{-0.2}\left(d_1, d_3; 1, 0.3\right); 1, 1\right), d_5; 0.6, 1\right)$$

$$\mu_p = \mathcal{P}_0\left(\mathcal{P}_{-\infty}\left(f_1, f_2; 1, 1\right), f_3; 1, 0.7\right)$$

Normalized weights:

ω_{d_1}	ω_{d_2}	ω_{d_3}	ω_{d_4}	ω_{d_5}	ω_{f_1}	ω_{f_2}	ω_{f_3}
3.3	2	1	3.3	16.1	9.9	9.9	13.8

Some results:

$(d_1, d_2, d_3, d_4, d_5)$	K_T / μ_p	K_B / μ_p	m / μ_p	μ_o
$(1.0, 0.9, 0.9, 1.0, 50)$	2832	5836	147	0.44
	0.23	0.14	0.62	(*)
$(1.1, 1.3, 1.2, 1.4, 100)$	3365	6029	170	0
	0.77	0.25	0	
$(0.7, 0.9, 0.8, 1.0, 0)$	2803	5730	144	0.34
	0.20	0.08	0.78	

minimum assumption:

$(1.1, 0.9, 0.8, 1.2, 150)$	2869	5933	156	0.2

minimum/geometric mean, equal weights:

$(0.9, 1, 0.9, 1.2, 50)$	2901	5876	156	0.36
$(0.7, 0.9, 0.9, 1.3, 50)$	2912	5820	157	0.36

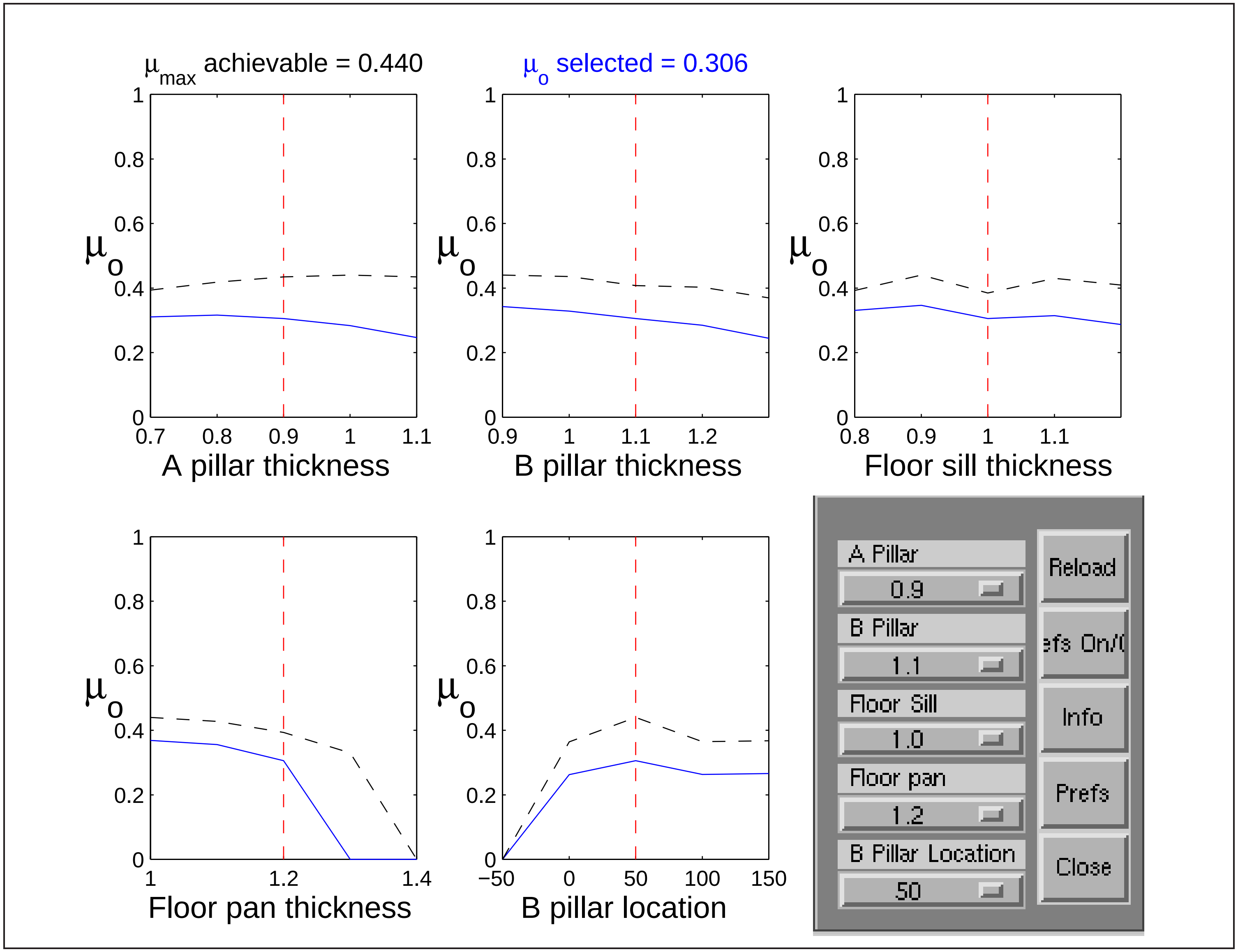
Approximations

These results come from a coarse but complete search ($5^5 = 3125$ runs, or about 50 hours). But:

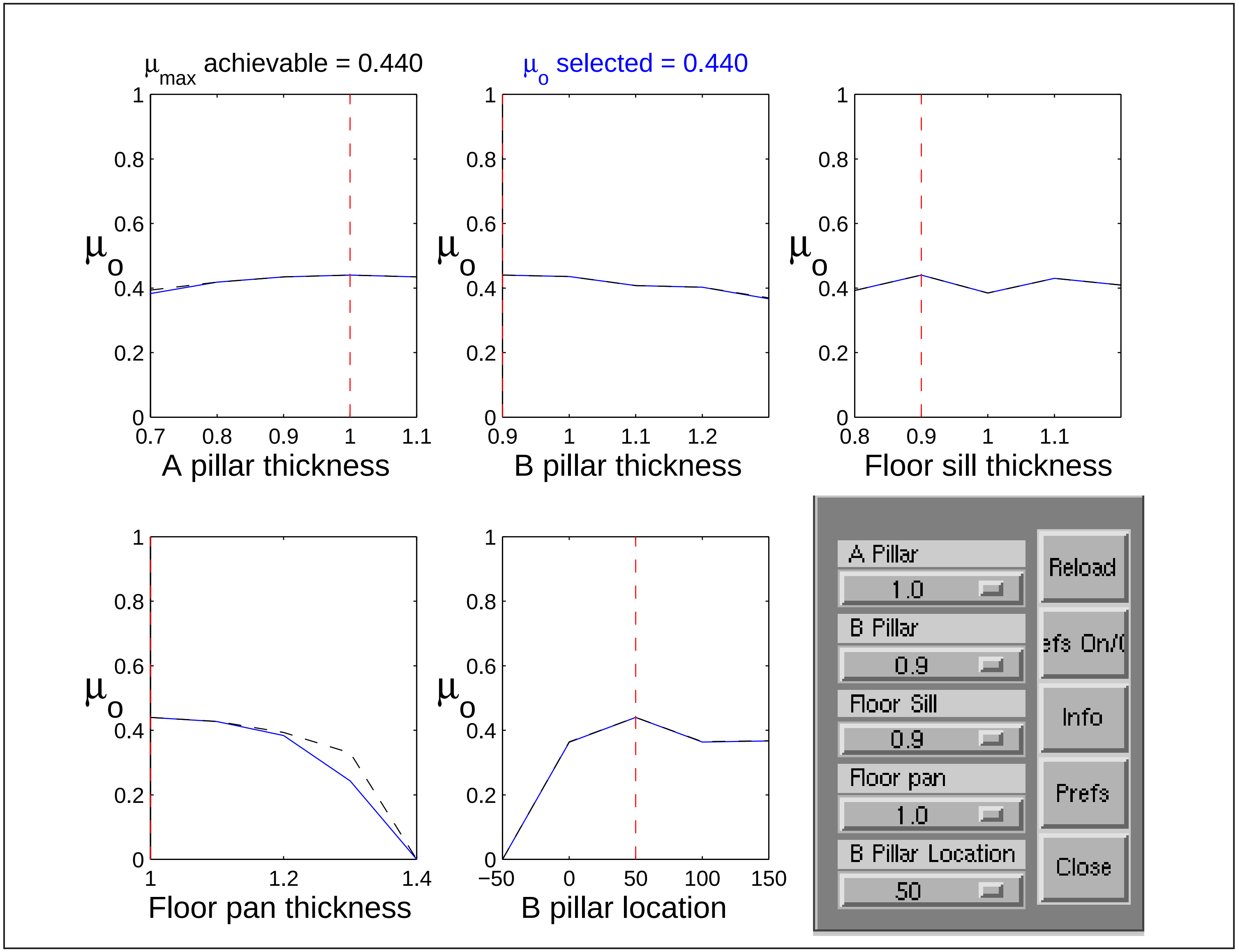
- At 5 minutes a run, that would be 11 days.
- At 7 points per axis, 1 minute a run, that's 12 days. [Miller, 1965]
- At 7 points per axis, 5 minutes a run, that's 58 days.

Approximation (with DOE) gives results within 4% in only 21 runs, or about 20 minutes.

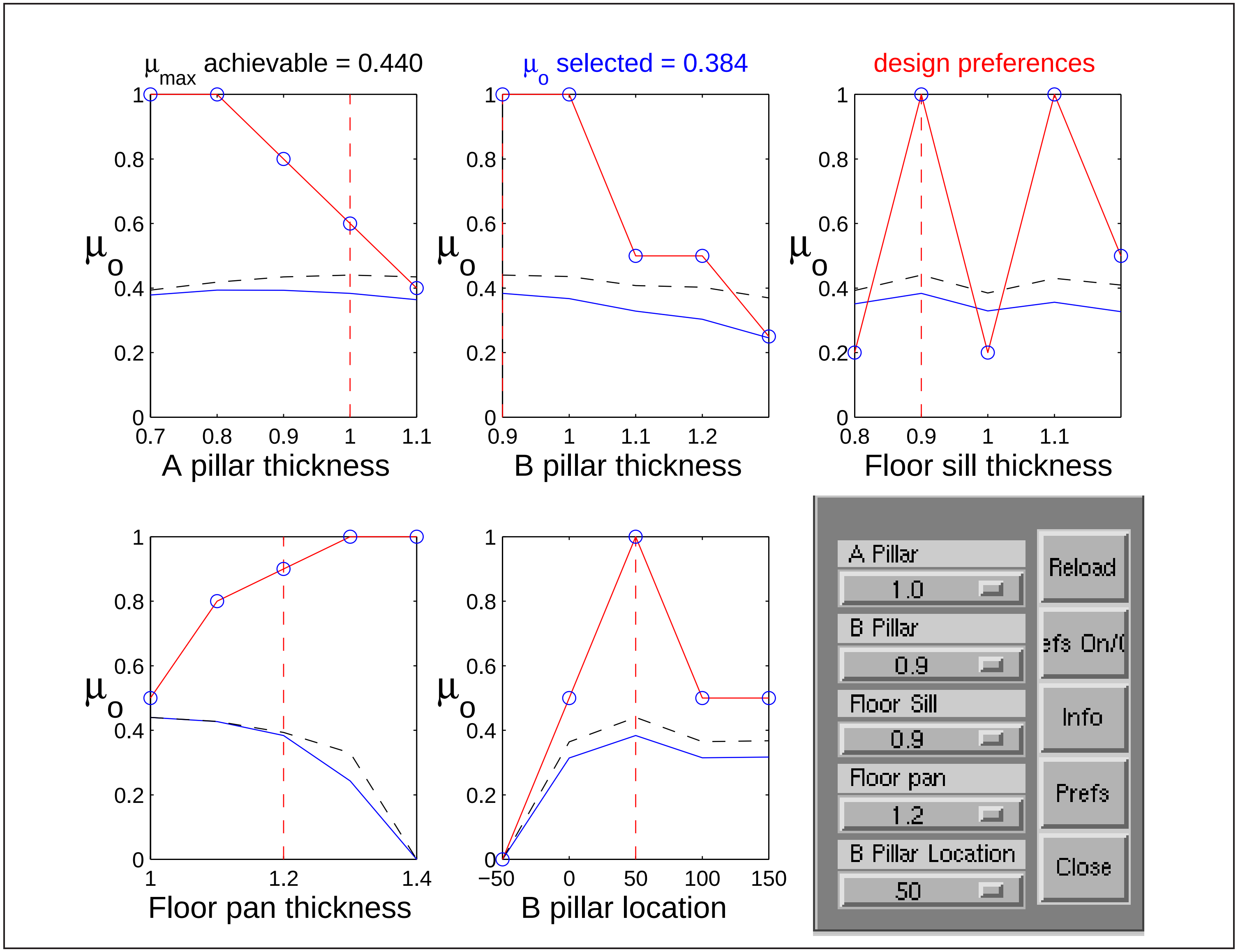
Note that recalculation with new strategies and weights is of negligible cost.



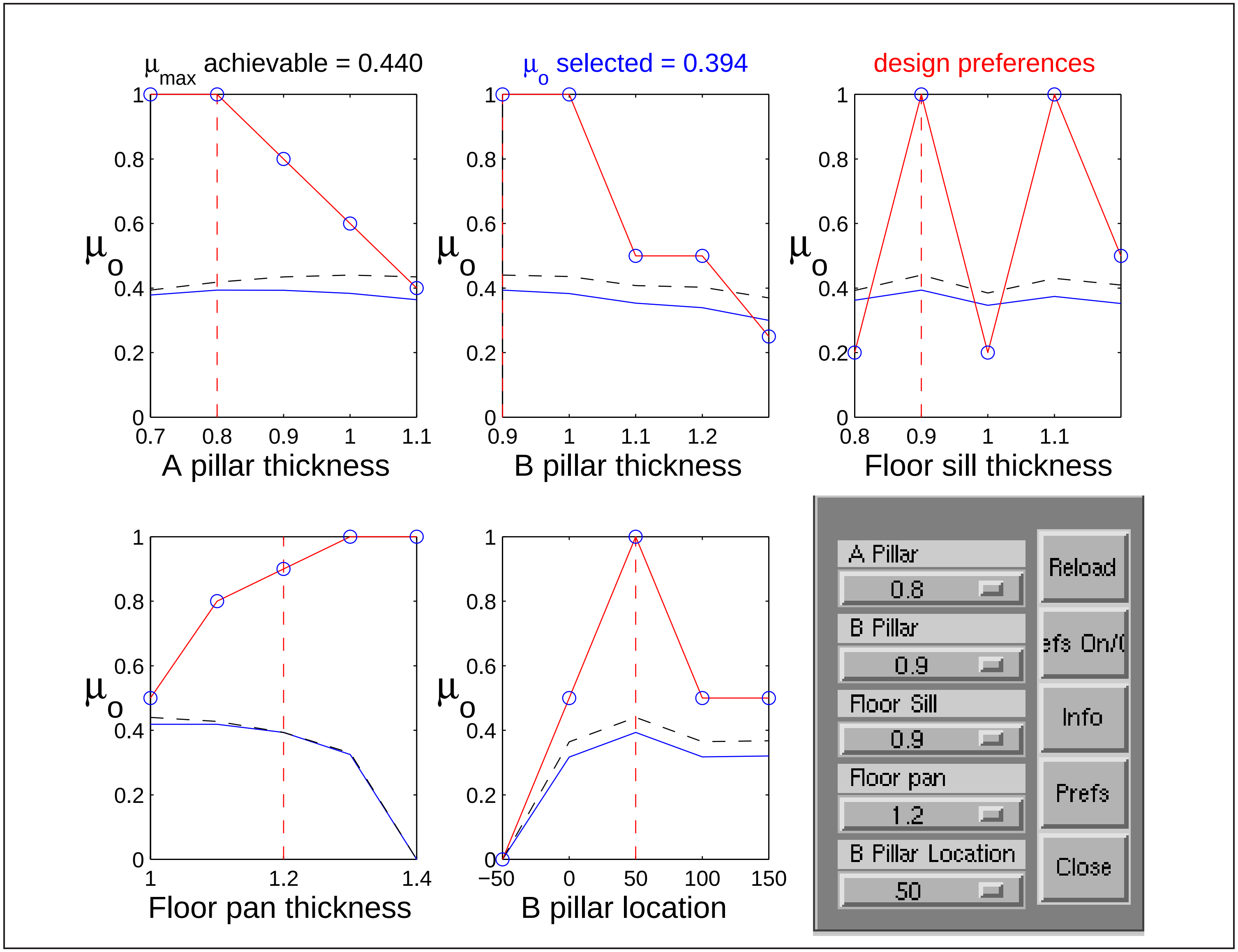
μmax achievable = 0.440
μo selected = 0.306
A pillar thickness
B pillar thickness
Floor sill thickness
Floor pan thickness
B pillar location
A Pillar
0.9
B Pillar
1.1
Floor Sill
1.0
Floor pan
1.2
B Pillar Location
50
Reload
Info
Prefs
Close

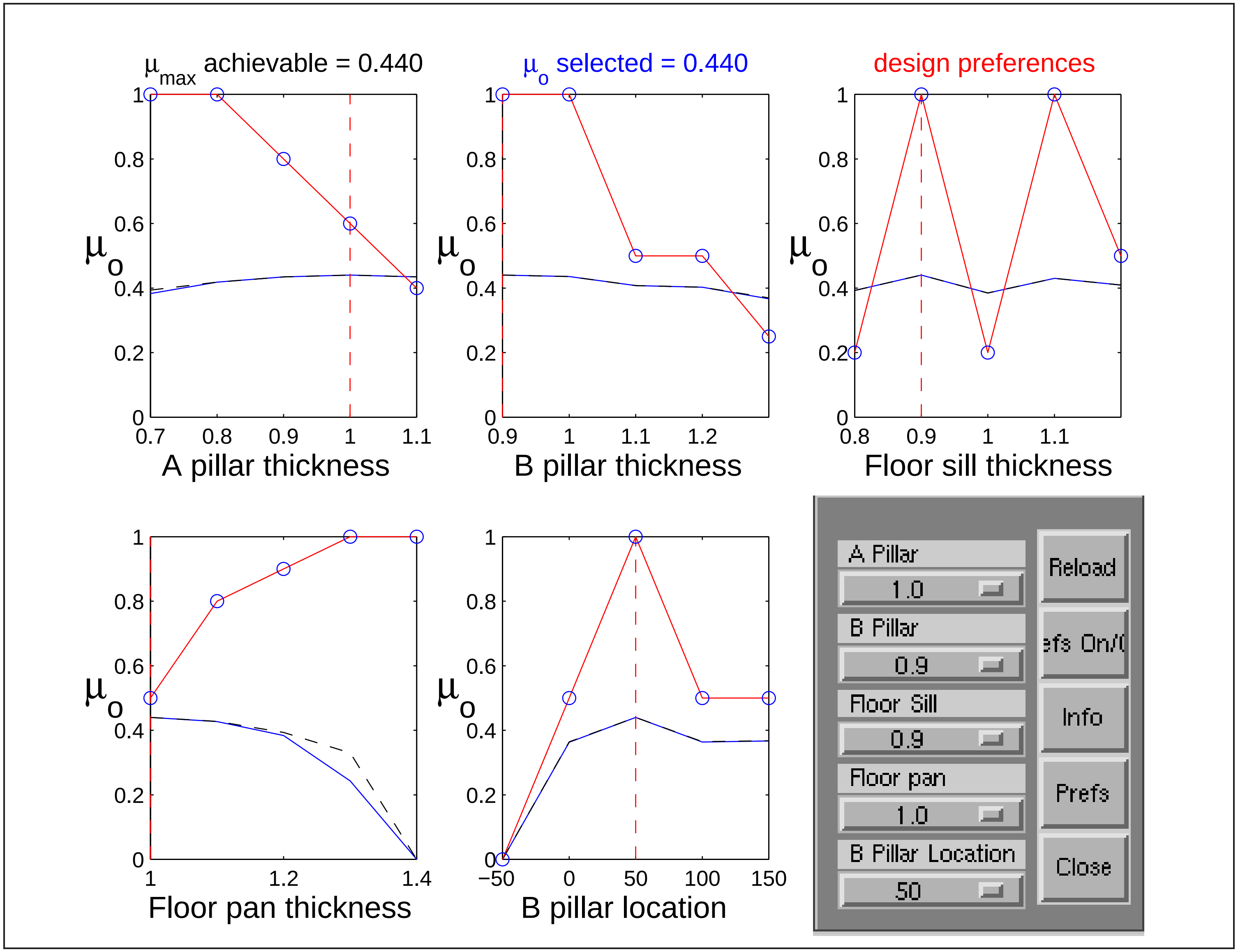


μmax achievable = 0.440
μo selected = 0.440
μo
A pillar thickness
B pillar thickness
Floor sill thickness
Floor pan thickness
B pillar location
A Pillar
1.0
B Pillar
0.9
Floor Sill
0.9
Floor pan
1.0
B Pillar Location
50
Reload
efs On/(
Info
Prefs
Close

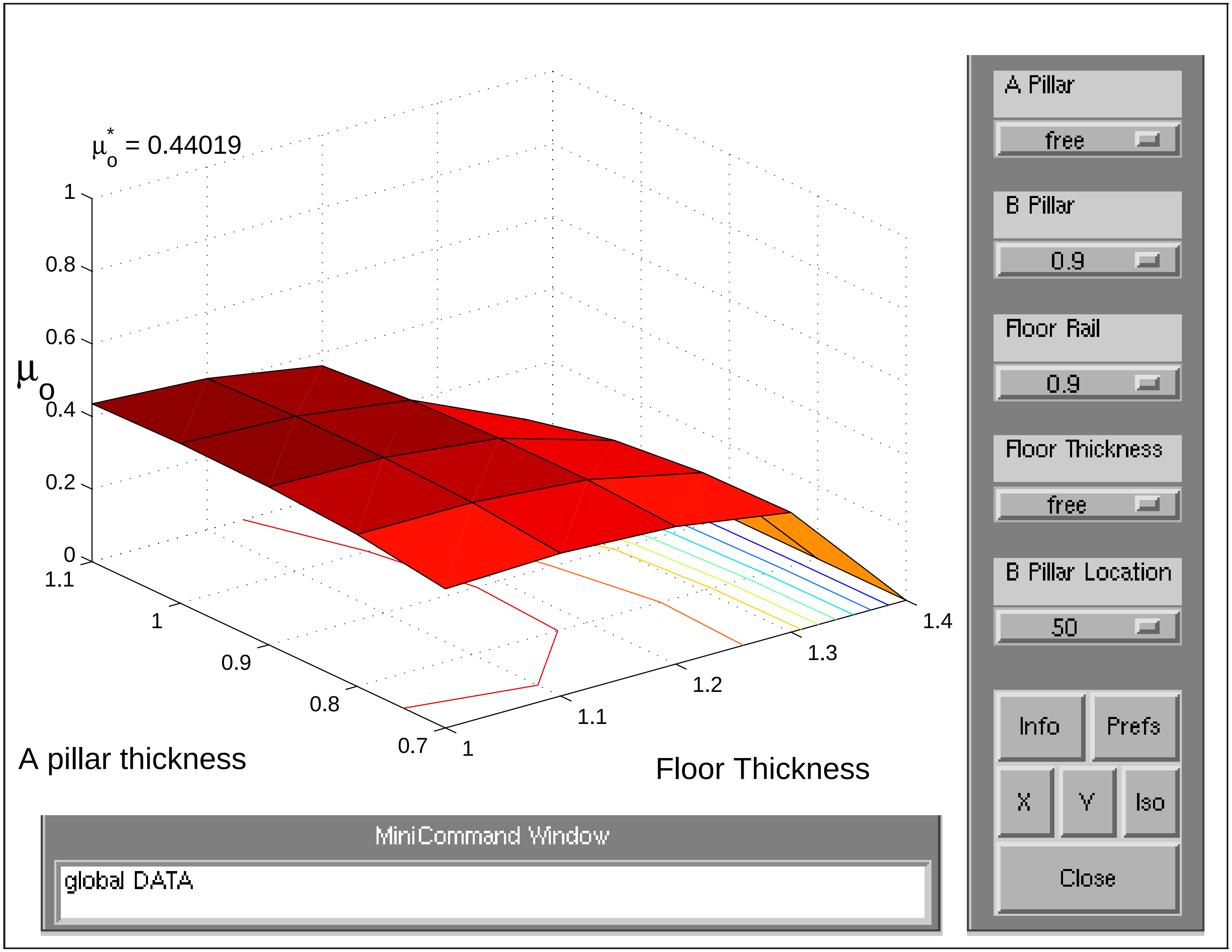


μmax achievable = 0.440
μo selected = 0.384
design preferences
μo
A pillar thickness
B pillar thickness
Floor sill thickness
Floor pan thickness
B pillar location
A Pillar
1.0
B Pillar
0.9
Floor Sill
0.9
Floor pan
1.2
B Pillar Location
50
Reload
efs On/(
Info
Prefs
Close

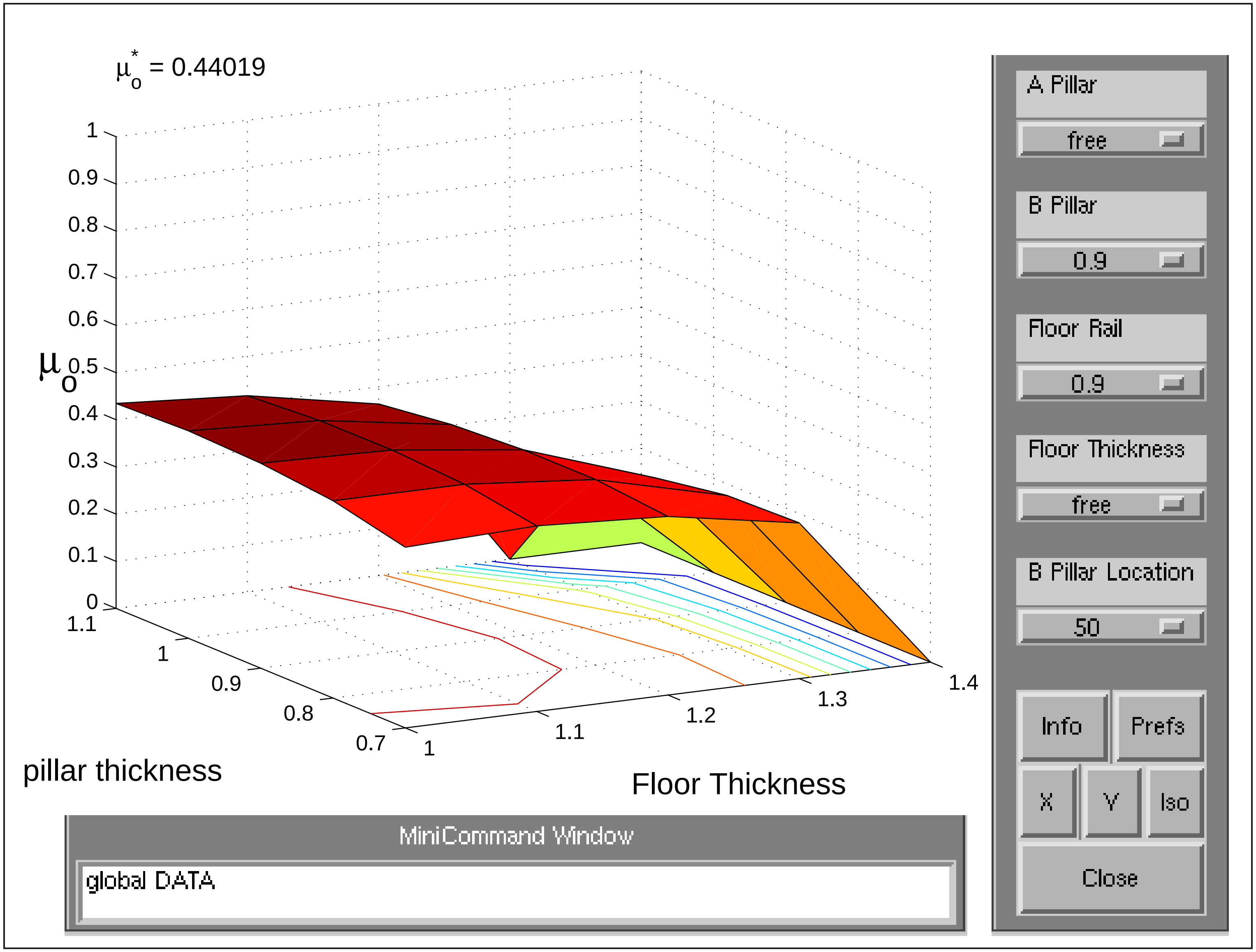

μmax achievable = 0.440
μo selected = 0.394
design preferences
μo
A pillar thickness
B pillar thickness
Floor sill thickness
Floor pan thickness
B pillar location
A Pillar
0.8
B Pillar
0.9
Floor Sill
0.9
Floor pan
1.2
B Pillar Location
50
Reload
efs On/(
Info
Prefs
Close



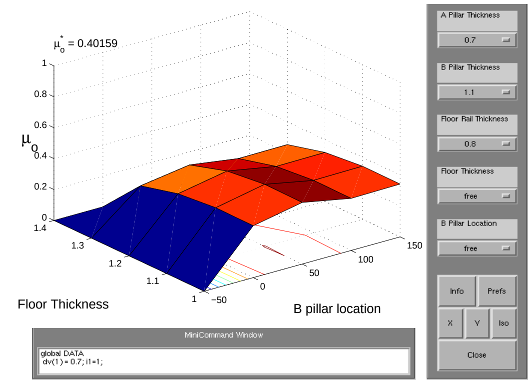
μ_{max} achievable = 0.440
μ_o selected = 0.440
design preferences
μ_o
A pillar thickness
B pillar thickness
Floor sill thickness
Floor pan thickness
B pillar location
A Pillar
1.0
B Pillar
0.9
Floor Sill
0.9
Floor pan
1.0
B Pillar Location
50
Reload
Info
Prefs
Close



$\mu_o^* = 0.44019$
μ_o
1
0.8
0.6
0.4
0.2
0
1.1
1
0.9
0.8
0.7
A pillar thickness
1
1.1
1.2
1.3
1.4
Floor Thickness
A Pillar
free
B Pillar
0.9
Floor Rail
0.9
Floor Thickness
free
B Pillar Location
50
Info
Prefs
X
Y
Iso
Close
MiniCommand Window
global DATA


$\mu_o^* = 0.44019$
μ_o
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1.1
1
0.9
0.8
0.7
1
1.1
1.2
1.3
1.4
pillar thickness
Floor Thickness
A Pillar
free
B Pillar
0.9
Floor Rail
0.9
Floor Thickness
free
B Pillar Location
50
Info
Prefs
X
Y
Iso
Close
MiniCommand Window
global DATA

Discussion of VW example



- Calculated overall preference μ_o depends on:
  - specified design preferences
  - specified performance preferences
  - performance analysis f (here, the FEM)
  - trade-off strategies and weights
- Trends:
  - overall performance varies with design preferences; exceptions are floor pan thickness (d_4) and A-pillar thickness (d_1).
  - styling dominates this decision problem (high weight).
- Choice of strategy affects the rank order of candidate designs.
- Explicit negotiation dependencies

Conclusions

Demonstration of a formalism for representing and manipulating imprecise descriptions of engineering designs, constraints and specifications that:

- incorporates uncomputed performance by direct specification of preferences.
- reconciles competing attributes explicitly and rationally.
- identifies and incorporates different trade-off strategies.
- effects of uncontrolled variations (noise) can be incorporated.
- set-based approach facilitates concurrency in design.
- provides a trade-off between computation cost and accuracy.

Now being applied to example design problems from industry.

A framework for making preferences, trade-offs and negotiations explicit:

- Decision *framework* rather than decision *making*.
- Support for iteration.
- Allows engineers to attach “soft” requirements (their own and others’) to an engineering model:
- Helps identify the rationale for decisions; can lessen political influence in design negotiations.
- Gives a graphical display of preference in many dimensions.

Sponsors:

- National Science Foundation
- NASA's Jet Propulsion Laboratory
- DARPA/NSF OPAAL Program