Multi-criteria exploration and decision making

Dr. Benny Raphael

Assistant Professor, Dept of Building, National University of Singapore

Overview

Introduction to multi-criteria decision making

Existing methods

- Weighting
- Outranking
- Pareto filtering

RR-PARETO3 algorithm and RRPExplorer tool

Conflicting objectives are ubiquitous

Example:

Maximize day light, minimize radiant heat inside a building

Example: Contractor selection for projects

Contractor	Bid price (million $)	Technical capability	Experience (years)	Financial status	Safety record	Management capability
1	**10**	Poor	4	Poor	Good	Poor
2	**12**	Good	40	Excellent	Average	Excellent
3	**12**	Excellent	20	Good	Average	Good
4	**13**	Poor	6	Good	Poor	Good

José Ramón San Cristóbal (2012) ‘Contractor Selection Using ulticriteria Decision-Making Methods’, JCEM, *ASCE, June, pp.751–758.*

Example: Select the best candidate for a PhD position

Name	Science	Design	Language	Prizes	Papers	Foreigner
Lee	**100**	60	40	2	2	No
Lam	**30**	100	70	3	0	No
Lin	**60**	70	70	1	0	No
Lim	**80**	50	90	1	1	No
Jim	**20**	10	30	3	0	Yes
Tim	**50**	60	50	0	0	Yes
Kim	**55**	55	45	1	0	Yes
Sim	**65**	60	55	1	0	No
Psy	**20**	45	90	3	0	Yes

How do you balance conflicting objectives?

Common approaches to handling multiple objectives

- Optimize the most important objective. Treat others as constraints (commonly used in designs based on codes of practice)

- Optimize a weighted sum of all the objectives (weighting and outranking methods)

- Pareto optimization

Weighting methods

A utility function or scoring function is used to combine the effects of all the criteria into a single number. Usually a weighted sum of all the criteria are taken. This score is used to rank alternatives

Examples:

- Price-Quality method for awarding tenders
- Cumulative Average Point (CAP) for grading students

Weighting methods

How do you assign weights?

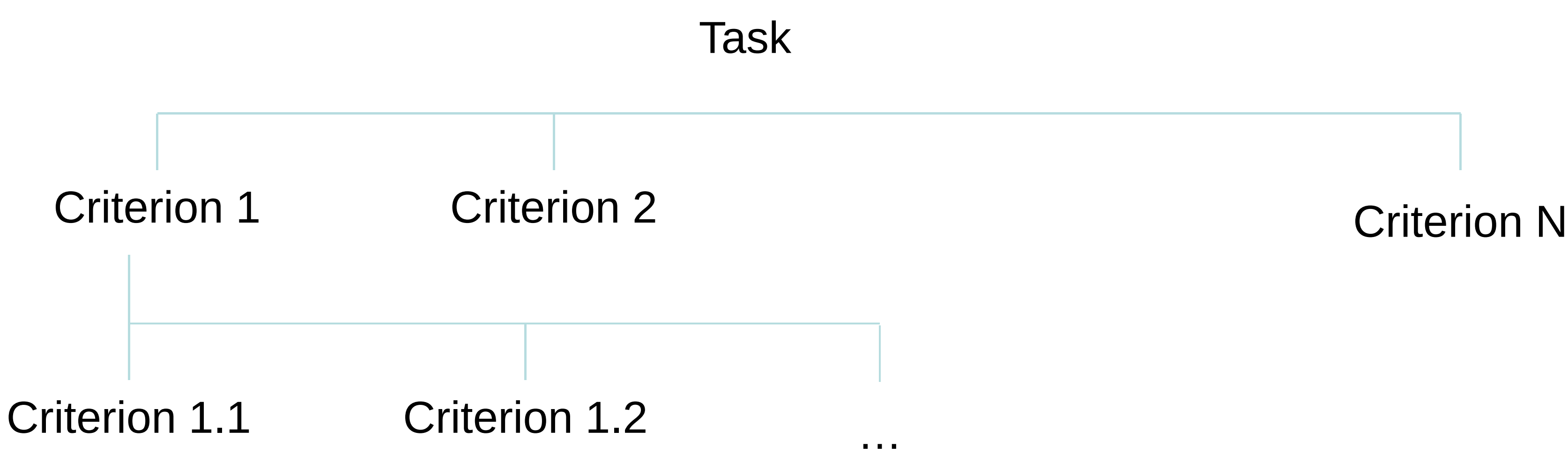
Weights represent preferences of the decision maker and are quite subjective. (It may not be easy to justify your assignment of weights to your boss!)

You can get any solution you want by adjusting the weights!

Analytic Hierarchy Process

1. Tasks are decomposed into a hierarchy of criteria

Saaty, T.L. (2008) 'Decision making with the analytic hierarchy process', *Int. J. Services Sciences, Vol. 1, No. 1, pp.83–98.*



2. Enumerate the alternatives (solutions)

Analytic Hierarchy Process

3. Use pair wise comparisons to bring out the relative importance of criteria. Pair wise scores are input manually by the user.

	Science	Design	Prizes
Science	1	2/1	4
Design	1/2	1	2/1
Prizes	1/4	1/2	1

1-9 Scale for comparing importance

Analytic Hierarchy Process

3. The first eigenvector of the matrix gives the weights for each criterion. Normalize it to get a total weight of 1.0

	weights
Science	0.57
Design	0.29
Prizes	0.14

Repeat for each sub-criteria

Analytic Hierarchy Process

4. Use pair wise comparisons to indicate the relative preference of each solution according to each criterion.

Papers	Lee	Lam	Lim
Lee	1	9	2
Lam	1/9	1	1/9
Lim	1/2	9	1

Analytic Hierarchy Process

4. The first eigenvector of the matrix gives the best assignment of scores for each solution according to this criterion

Normalize it to get

a total score of 1.0

	weights
Lee	0.58
Lam	0.05
Lim	0.36

Repeat for each sub-criteria

Analytic Hierarchy Process

5. Weighted sum of scores of sub-criteria give the score for the parent criterion and so on.

Task

Criterion 1
Score: 0.21
(0.4*0.1 + 0.1*0.7 + 0.5*0.2)

Criterion 2

Criterion N

Criterion 1.1
Score: 0.1
Weight: 0.4

Criterion 1.2
Score: 0.7
Weight: 0.1

Criterion 1.3
Score: 0.2
Weight: 0.5

Analytic Hierarchy Process

The total score is the weighted sum of the scores of the top level criteria.

	Total score
Lee	0.58
Lam	0.05
Lim	0.36

The winner has the highest total score.

Analytic Hierarchy Process

Summary

- A systematic approach to manual decision making using pair wise comparisons of criteria and solutions. Relies on expert opinion for comparisons.
- Accommodates both qualitative and quantitative criteria.
- Drawback: Large number of pair wise comparisons!

A variation of the approach called Analytic Network Process is also popular

PROMETHEE

Preference Ranking Organization METHod for Enrichment Evaluations: Features

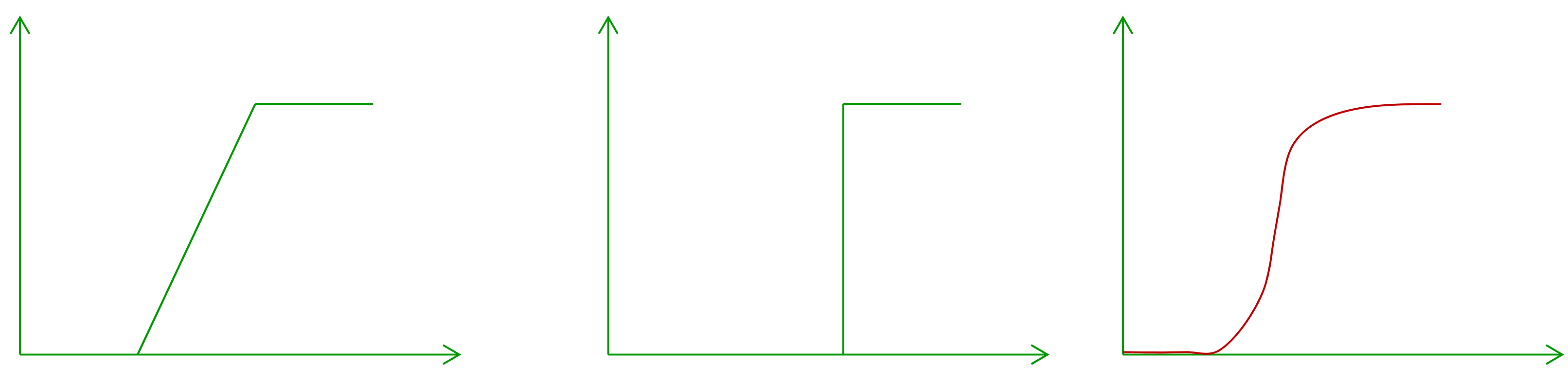
- An “Outranking” method using pair-wise comparisons of solutions according to each criterion.
- Differences between solutions according to each criterion is converted to a scale [0-1] using preference functions
- Weights associated with each criterion (equal weights by default)

Brans J.P. et Mareschal B. Chapter 5, “Promethee Methods” in Multiple Criteria Decision Analysis: State-of-the-art Surveys, Eds. Figueira, J., Greco, S. and Ehrgott, M. Springer, 2004, pp 164-195.

PROMETHEE

Preference Function

- Compute the deviation d_j (a,b) between two solutions (a,b) for a given criterion (j)
- The preference function P_i (a,b) gives a value from 0 to 1 depending on the value of d_j (a,b). It indicates the degree to which solution *a* is preferred over *b*.
- Possible to include thresholds of preference



PROMETHEE

Aggregated Preference Indices:

Weighted sum of preferences over all criteria

$$\Pi(a,b) = \sum_{j=1}^{N} P_j(a,b) w_j$$

$$\Pi(b,a) = \sum_{j=1}^{N} P_j(b,a) w_j$$

PROMETHEE

Outranking flows for a solution:

Average preferences over all other solutions. Net outranking flow is used to order the solutions

Positive outranking flow

$$\Theta^{+}(a) = \frac{1}{k-1}\sum_{x} P_j(a, x)$$

Negative outranking flow

$$\Theta^{-}(a) = \frac{1}{k-1}\sum_{x} P_j(x, a)$$

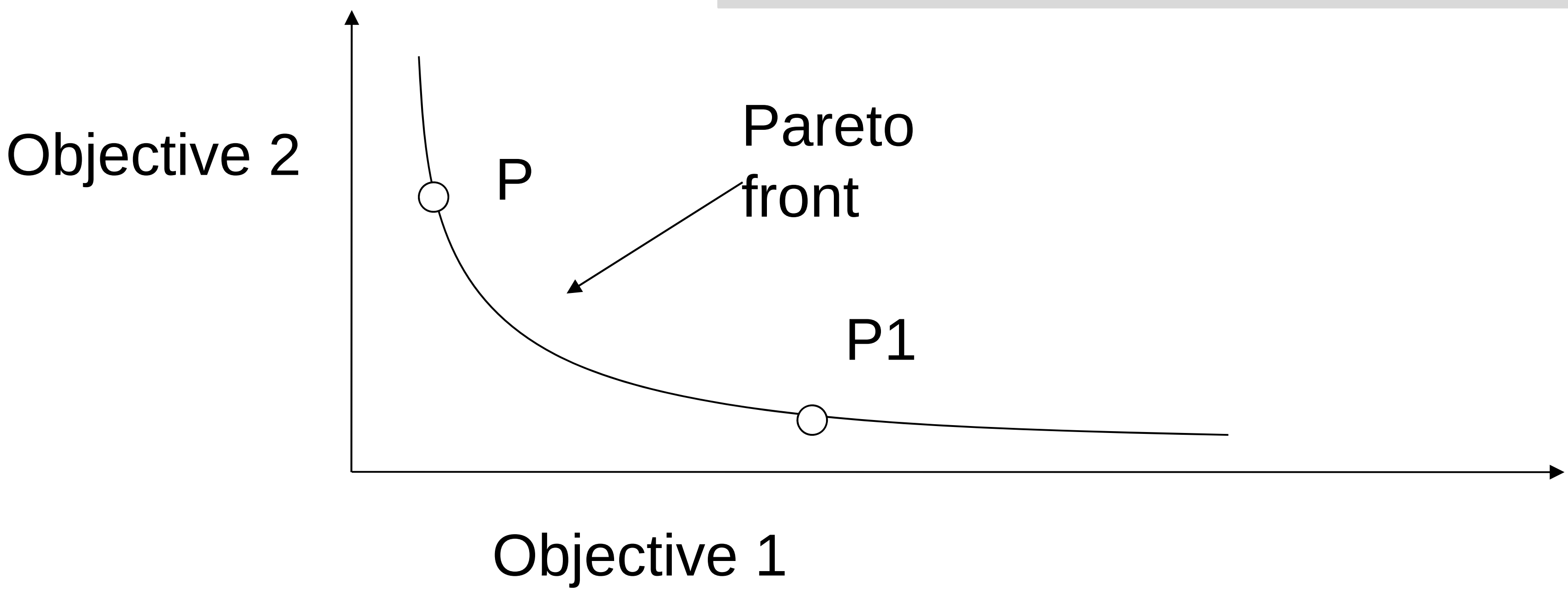
Net outranking flow

$$\Theta(a) = \Theta^{+}(a) - \Theta^{-}(a)$$

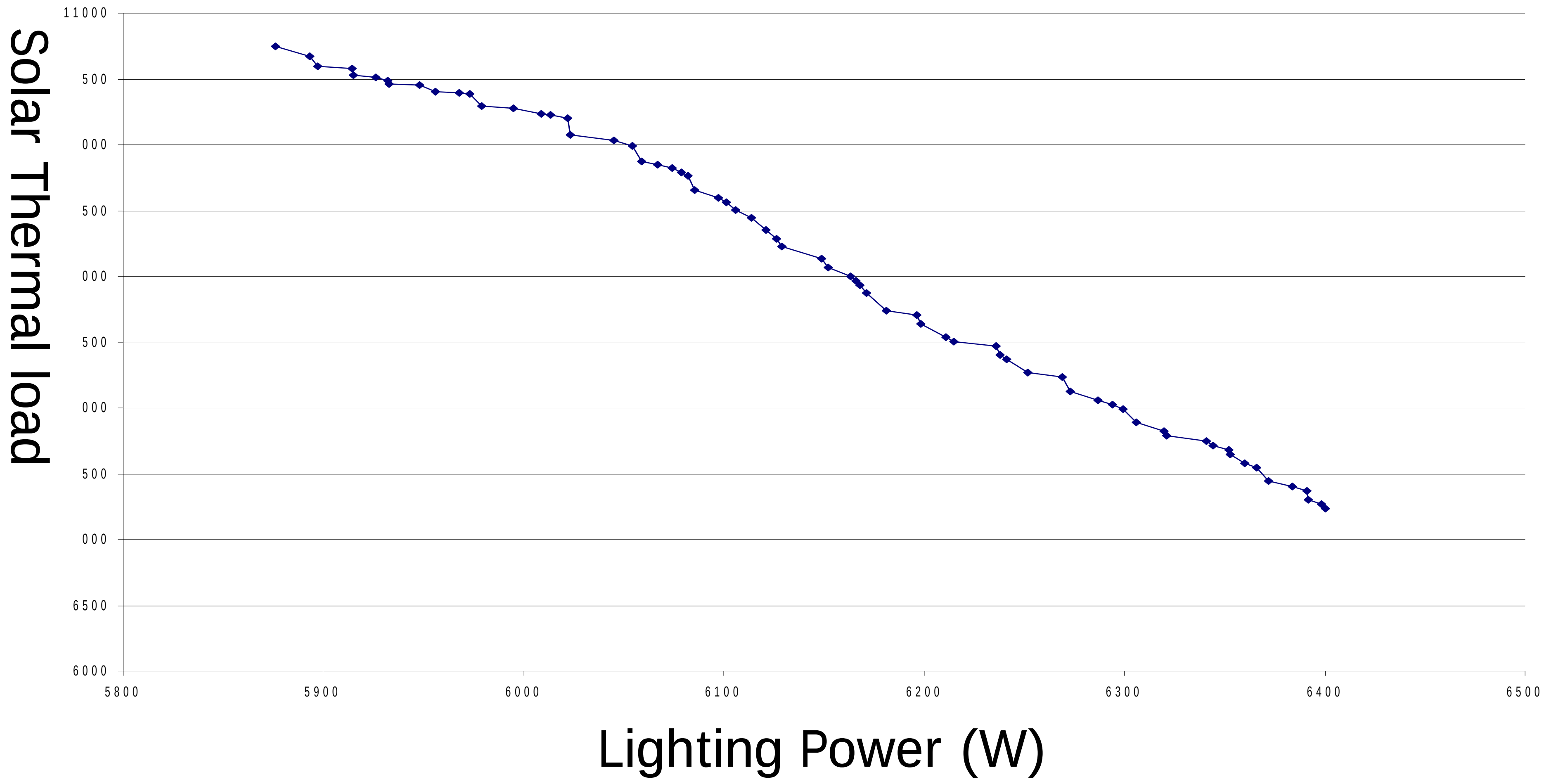
Pareto filtering

A solution point P is accepted only if there are no solutions better than P with respect to all the objectives

Raphael, B. and I.F.C. Smith, Fundamentals of computer aided engineering, John Wiley, 2003



Pareto front



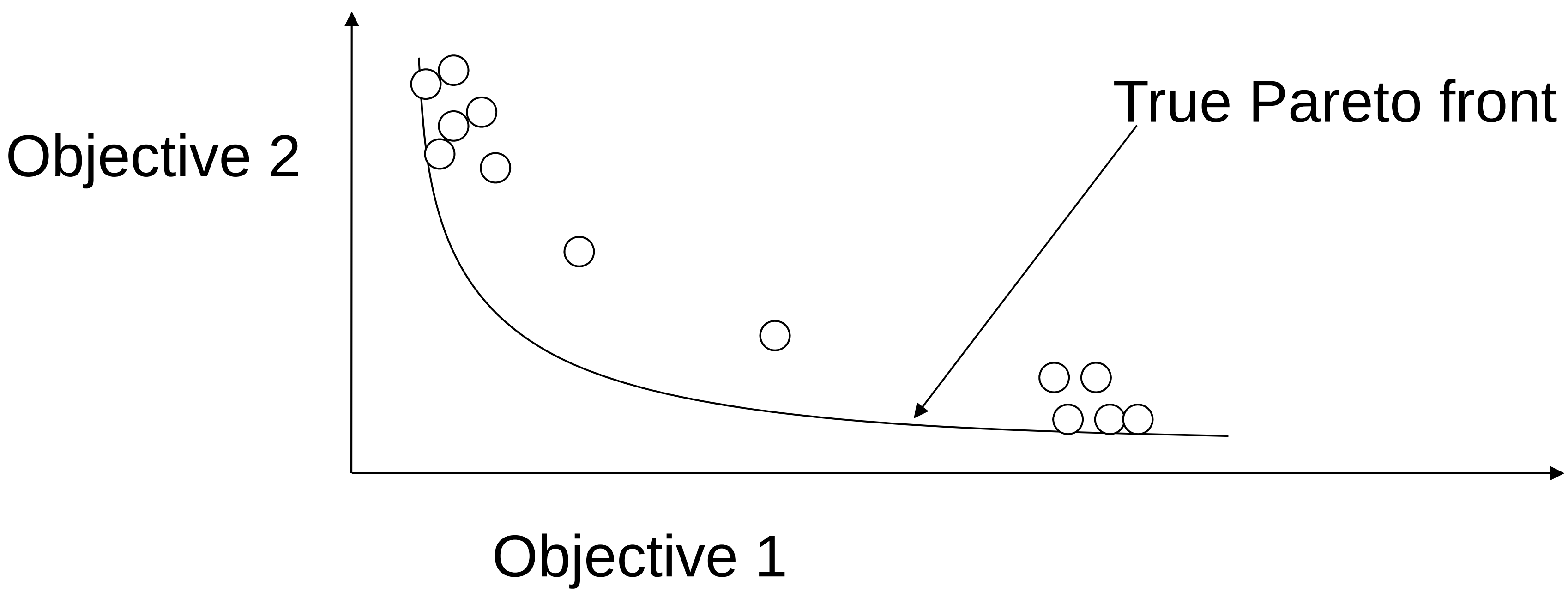
Generating Pareto Front

- Multi-start methods
- Multi-objective versions of global search methods such as Genetic Algorithms, Simulated Annealing and PGSL
- Weighting methods

No mathematical proof exists that these methods converge to the true Pareto optimal front

Characteristics of a good Pareto Front

- Many points should be generated
- The points should be uniformly distributed.
- The points should be close to the theoretical Pareto front.



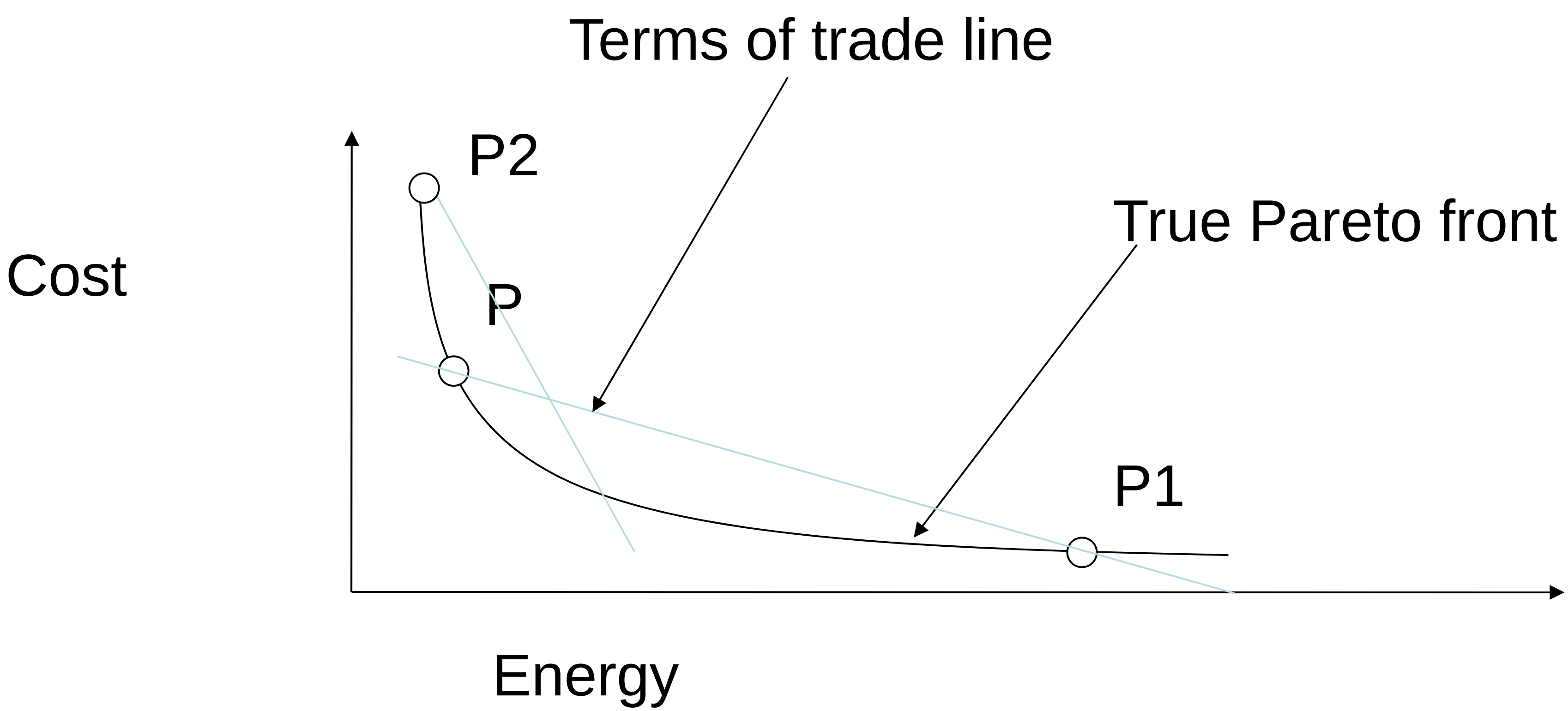
Difficulties with the Pareto approach

- No support to select a single best solution
- In control tasks, you cannot ask the user to select the best action all the time!
- The efficient front changes completely if you add a new constraint

PEG: Pareto Edgeworth Grierson

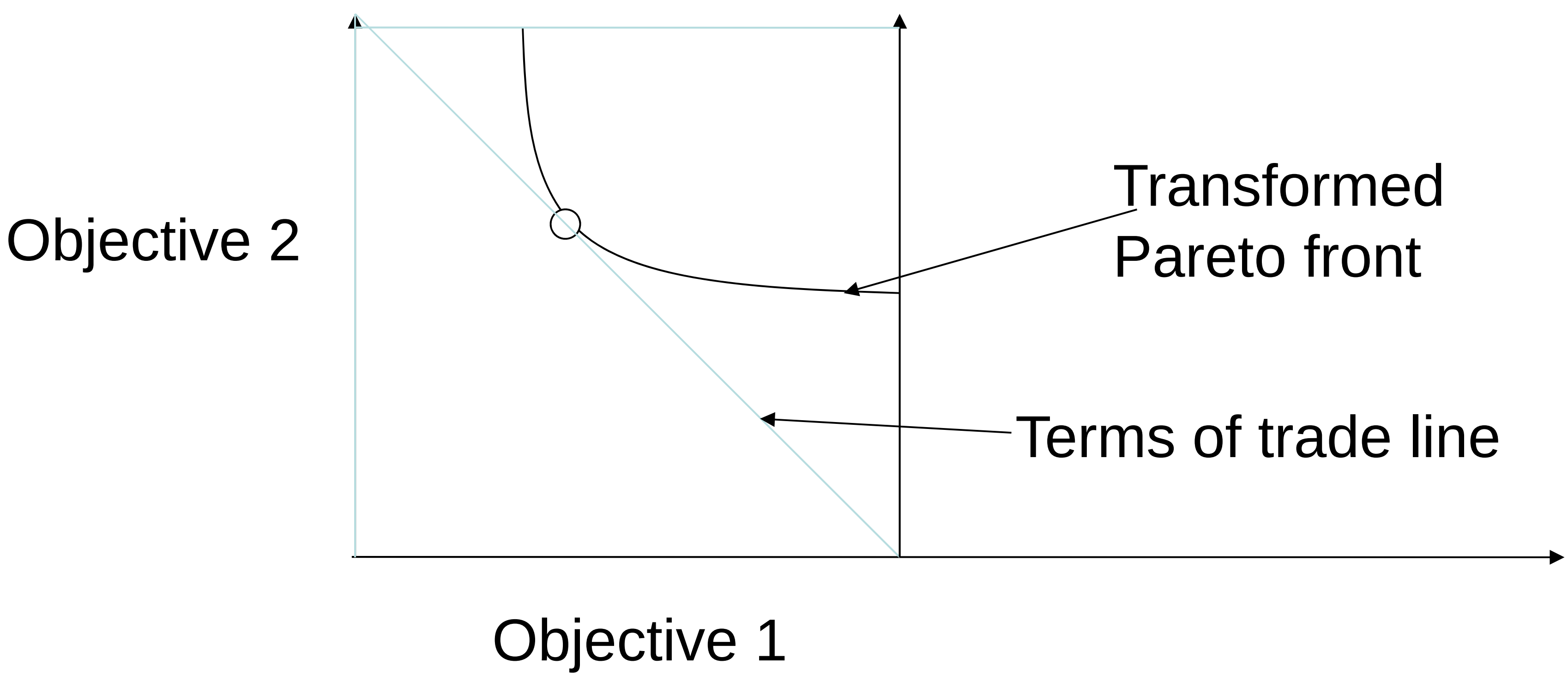
Donald E. Grierson (2008) ‘Pareto multi-criteria decision making’, *Advanced Engineering Informatics, 22, pp. 371–384.*

Choose the same trade-off with respect to all criteria



PEG: Pareto Edgeworth Grierson

Optimal solution is where the diagonal of the PEG cube touches the transformed Pareto front



Other multi-criteria methods

- ELECTRE: Elimination and choice expressing reality
- MAUT: Multi Attribute Utility Theory
  - Relationship between utility and costs
- Fuzzy set, Rough sets
  - Models uncertainty
- TOPSIS
  - Concept of positive and negative ideal solution

RR-PARETO3 Algorithm

The best compromise solution is selected using

- Ordering of objectives according to their importance
- Sensitivity of objectives

The sensitivity parameter captures information about how much increase in the objective function value is acceptable to the user

Raphael B., Multi-criteria decision making for collaborative design optimization of buildings, Built Environment Project and Asset Management, (Emerald publishers), Vol 1, Issue 2, Nov, 2011

Sensitivity of objectives

- Represents domain knowledge or expert view point or the judgment of the decision maker
- Are treated similar to a hard constraint by RR-PARETO3

If users have no strong opinion about what is acceptable, there is no need to specify the sensitivity. There is a default procedure for identifying a compromise solution

Two-stage filtering

Stage 1: Using ordering and sensitivity of objectives specified by users

Stage 2: Filtering out less attractive solutions by the method of bisection

Stage 2 filtering is performed if there is more than one solution after Stage 1.

Example of blind control

- Scenario 1:
  - Lighting load has higher priority
  - Sensitivity of both objectives 2%
- Scenario 2:
  - Sensitivity of both objectives 5%

Blind Position (m)	Lighting (W)	Solar (W)
0.1	6400	7234
0.2	6360	7580
0.51	6198	8637
0.6	6168	8936
0.73	6121	9351
1	6023	10079
1.15	5979	10295
1.24	5973	10388
1.25	5968	10397
1.26	5956	10406
1.31	5948	10452
1.32	5933	10461
1.35	5932	10488
1.38	5926	10515
1.4	5915	10531
1.5	5897	10596
1.56	5876	10744

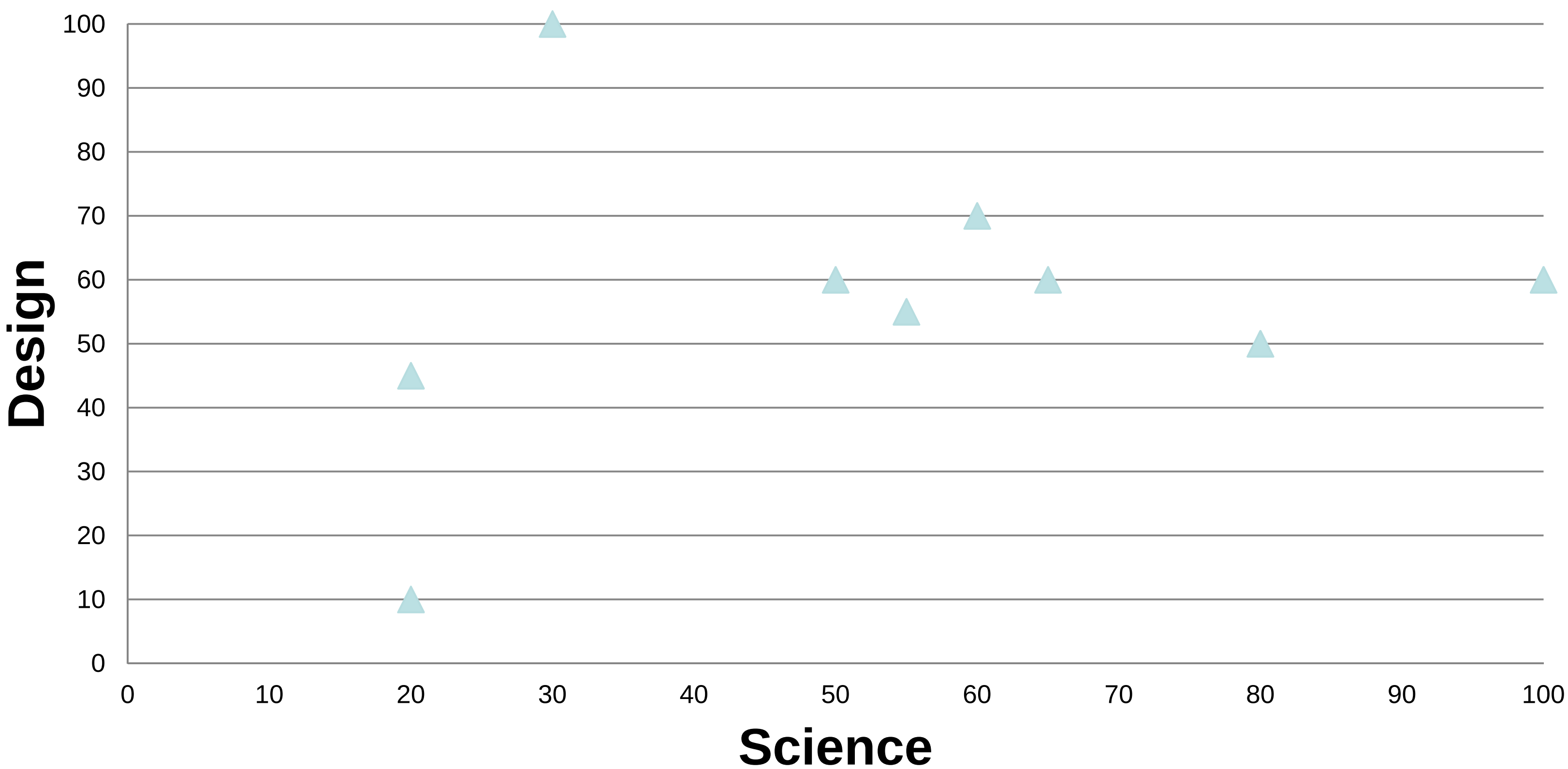
Blind Position (m)	Lighting (W)	Solar (W)
0.1	6400	7234
0.2	6360	7580
0.51	6198	8637
0.6	6168	8936
0.73	6121	9351
1	6023	10079
1.15	5979	10295
1.24	5973	10388
1.25	5968	10397
1.26	5956	10406
1.31	5948	10452
1.32	5933	10461
1.35	5932	10488
1.38	5926	10515
1.4	5915	10531
1.5	5897	10596
1.56	5876	10744

Blind Position (m)	Lighting (W)	Solar (W)
0.1	6400	7234
0.2	6360	7580
0.51	6198	8637
0.6	6168	8936
0.73	6121	9351
1	6023	10079
1.15	5979	10295
1.24	5973	10388
1.25	5968	10397
1.26	5956	10406
1.31	5948	10452
1.32	5933	10461
1.35	5932	10488
1.38	5926	10515
1.4	5915	10531
1.5	5897	10596
1.56	5876	10744

Stage 2 filtering

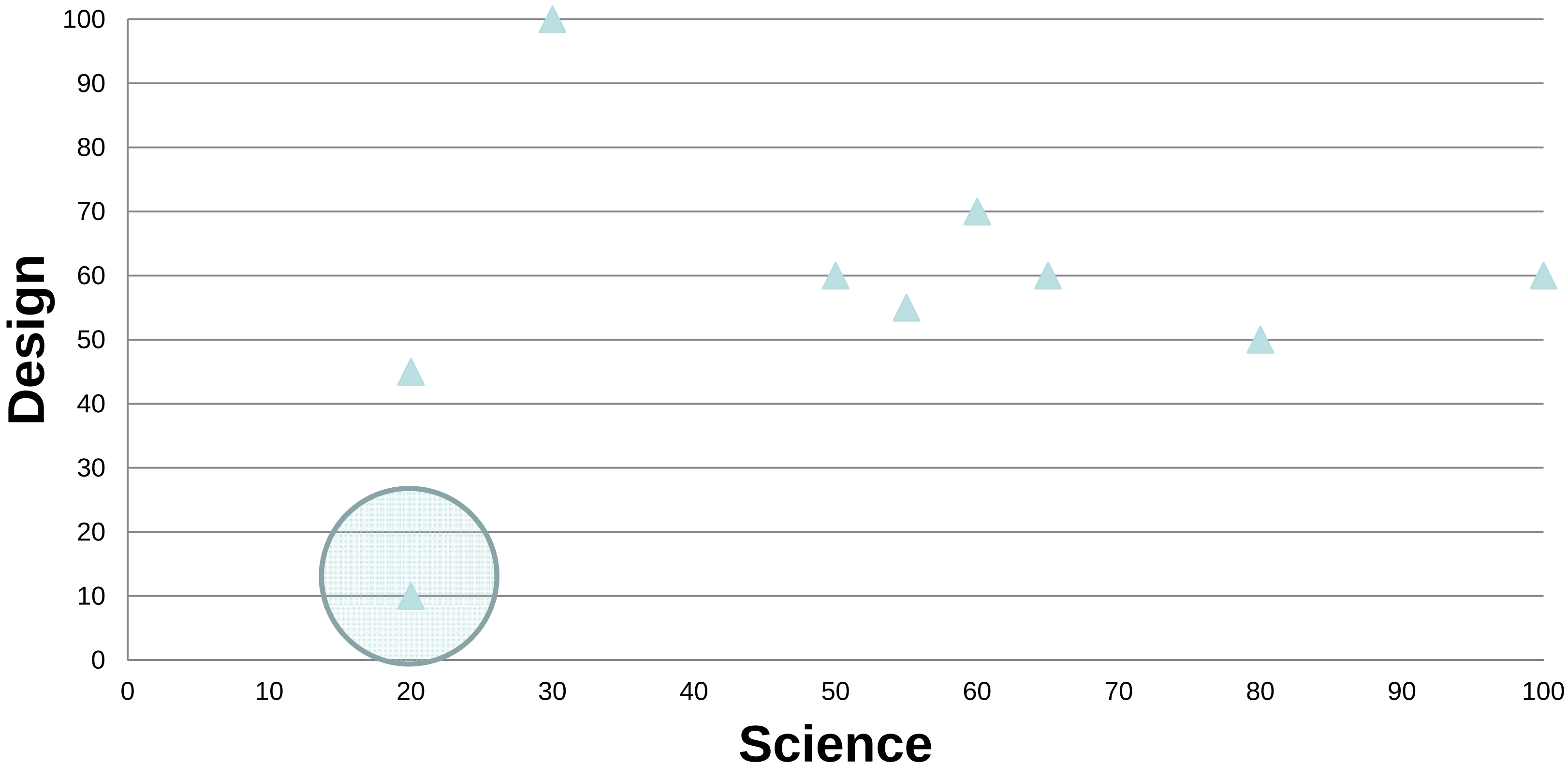
What should be the default procedure when no sensitivity is specified?

How do you determine what is the best trade-off?



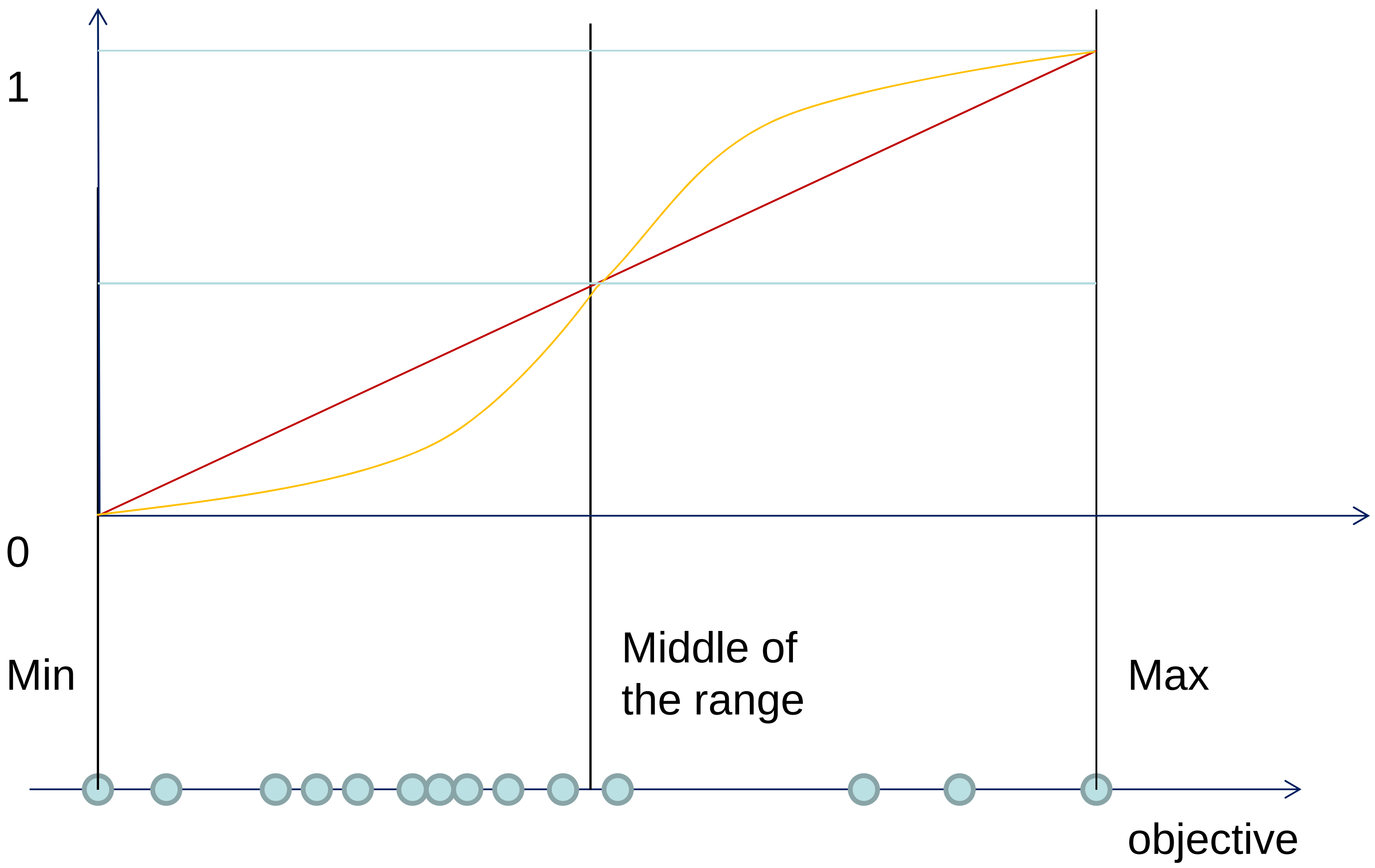
Stage 2 filtering

People might disagree on what is the best trade-off and which is the best compromise solution. But usually there is a consensus on clearly identifiable bad solution!



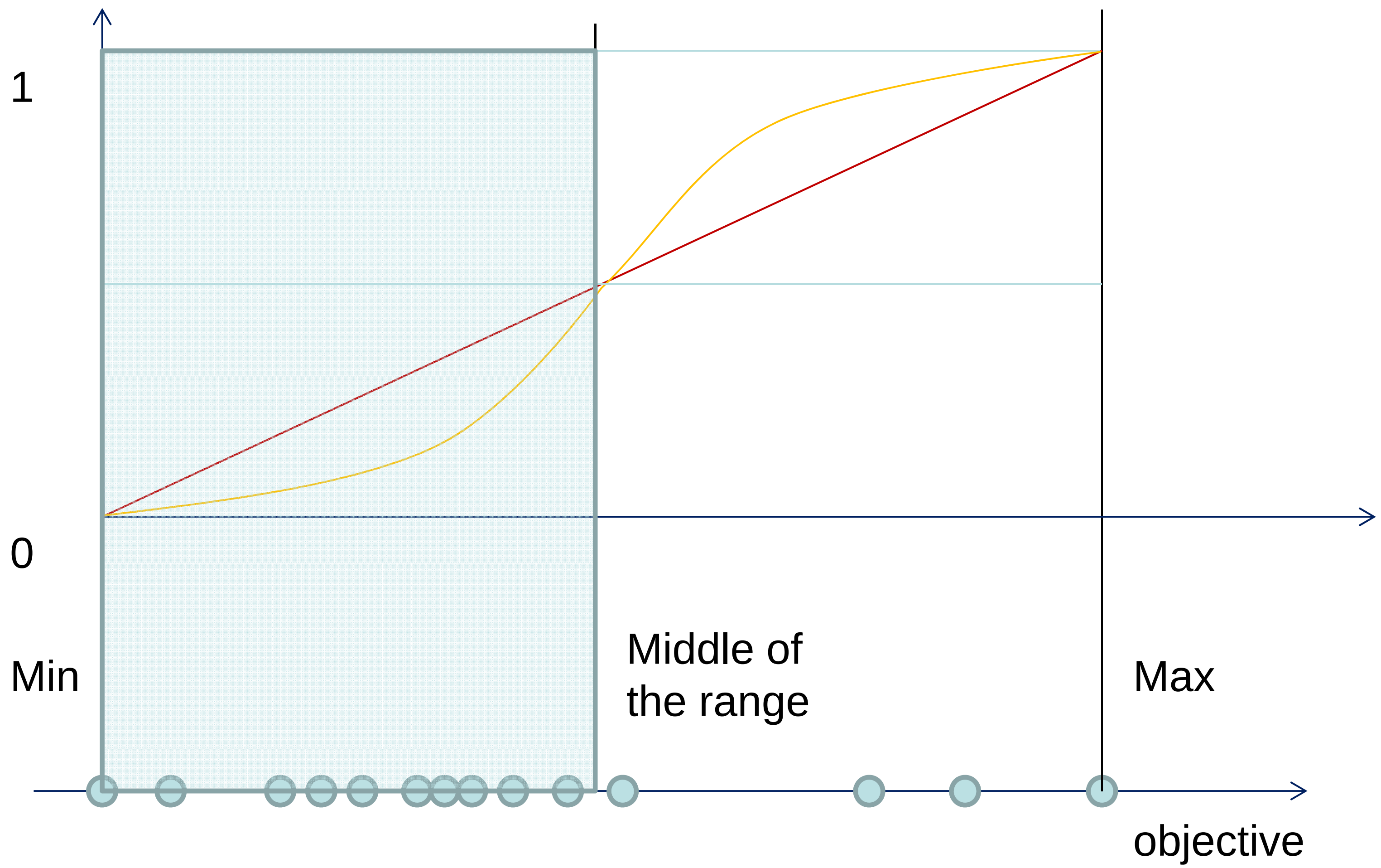
Scale of goodness

Acceptable range should not depend on the distribution or clustering of points!



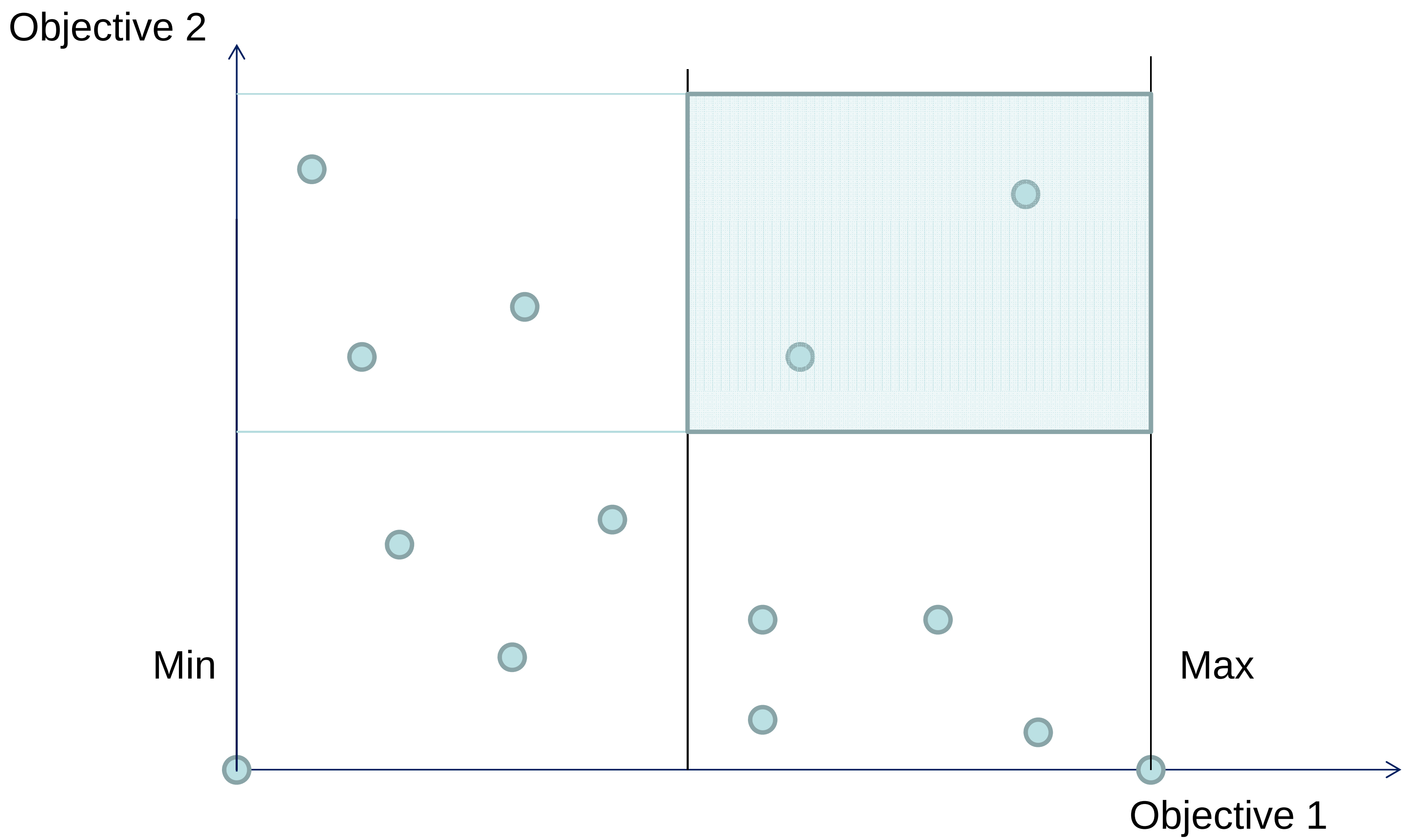
Method of bisecting the hypercube

Values below the middle of the range are considered to be no good.
(Remember! We want to select good solutions near the extreme end)



Method of bisecting the hypercube

A “Relaxed” form of Pareto filtering?



Stage 2 filtering

Aims to eliminate the worst solutions according to this heuristic

- Stage 2.1: Points that lie in the bad half of most objectives are eliminated (taking combinations of k most important objectives at a time, k = N to 2)
- Stage 2.2: Points that lie in the bad half of individual objectives are eliminated according to the order of importance of objectives
- Stage 2.3: Iteratively remove the worst point according to each objective based on the order of importance

RR-PARETO: “Restricting” to regions on the “Relaxed” Pareto front where there are no “no-good” solutions

Stage 2 filtering

The bisection of hypercube helps to remove visibly obvious bad solutions!

When there are many options, each objective is given a chance to remove the worst candidate. The most important objective is given the first chance.

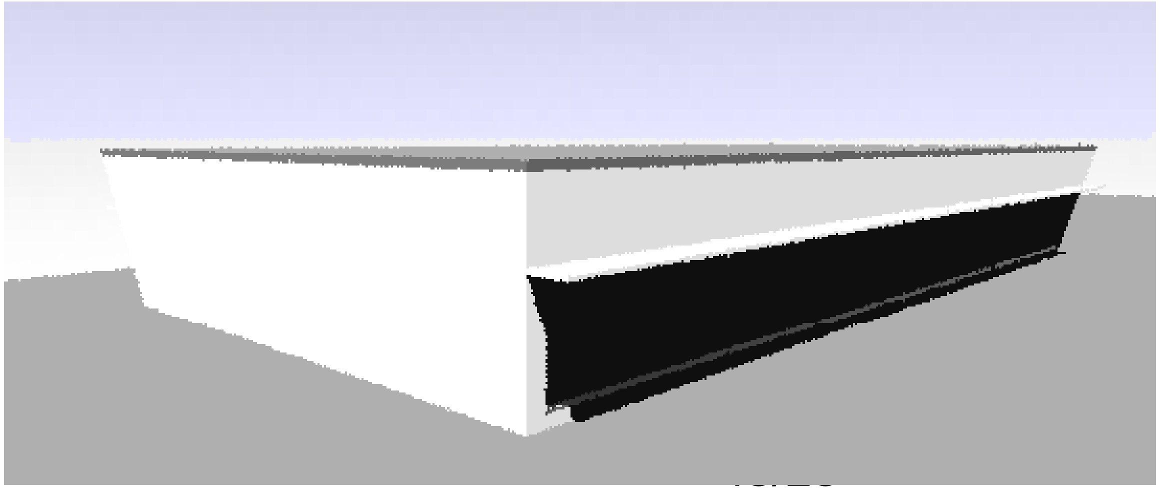
Stages 2.1 and 2.2 use values of objective function for elimination, while Stage 2.3 uses ranking of solutions

The filtering algorithm was empirically tested on a number of data in order to find out if it generates solutions that can be justified using simple reasoning that human users can easily accept.

Application example

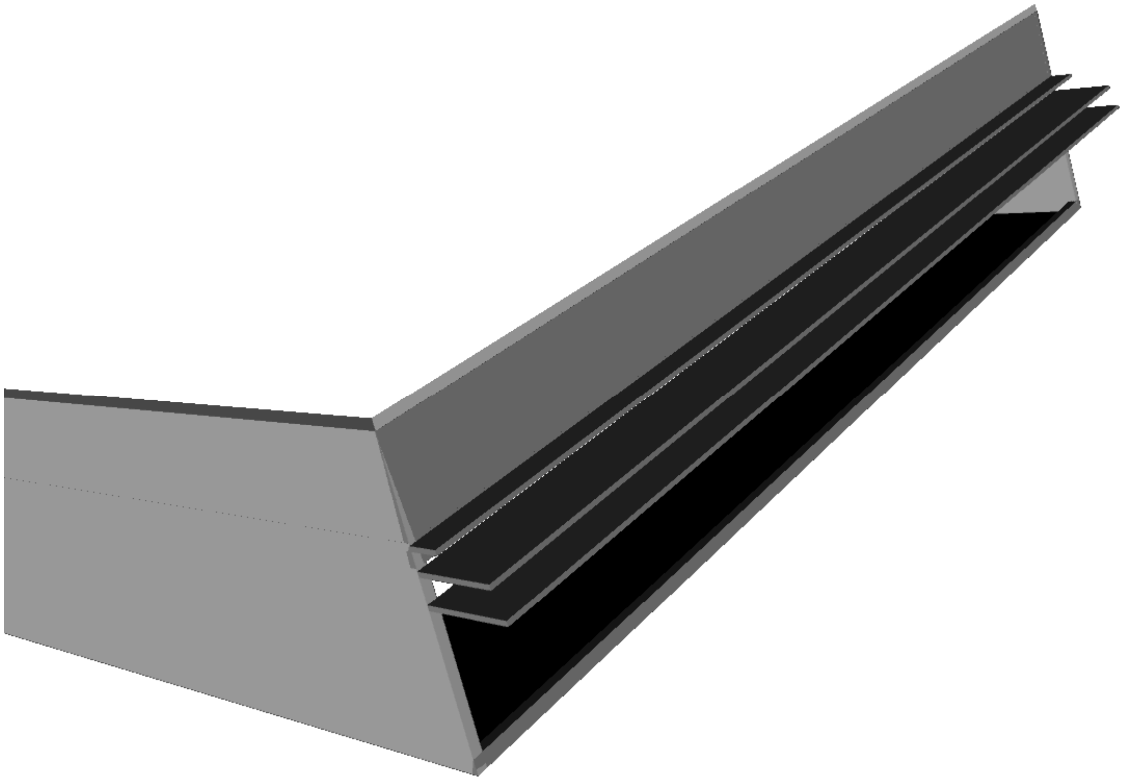
Applied to a blind control application

- The optimal blind positions are computed for each hour of the day
- Energy savings is computed by comparing with a traditional control strategy

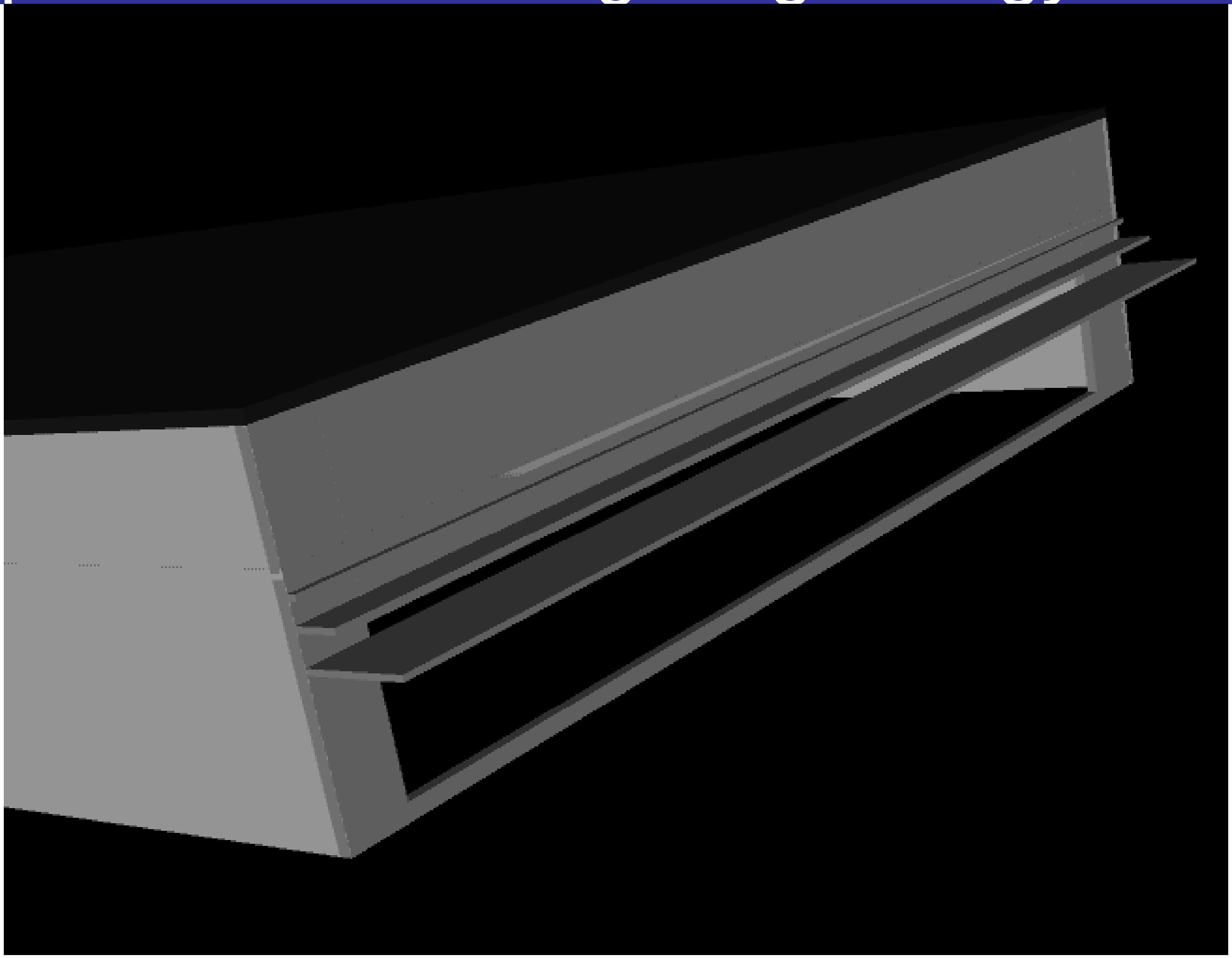


Hour	Blind position		Lighting (KW)	Cooling (KW)	Total Energy (KWH)	
	W1	W3			Integrated Control Strategy	Control strategy 2
8	100	63	5.52	11.09	20.57	29.5
9	49	45	5.91	11.48	21.35	30.08
10	26	39	6.06	11.73	21.75	30.47
11	100	40	5.74	12.28	21.97	30.61
12	100	78	5.45	12.59	22	30.52
13	100	100	5.31	12.67	21.95	21.95
14	85	96	5.39	12.7	22.05	30.63
15	49	18	6.02	12.26	22.24	31
16	37	21	6.09	12.13	22.19	31.21
17	41	30	5.99	11.93	21.88	30.83
18	55	76	5.61	11.48	21.05	29.97

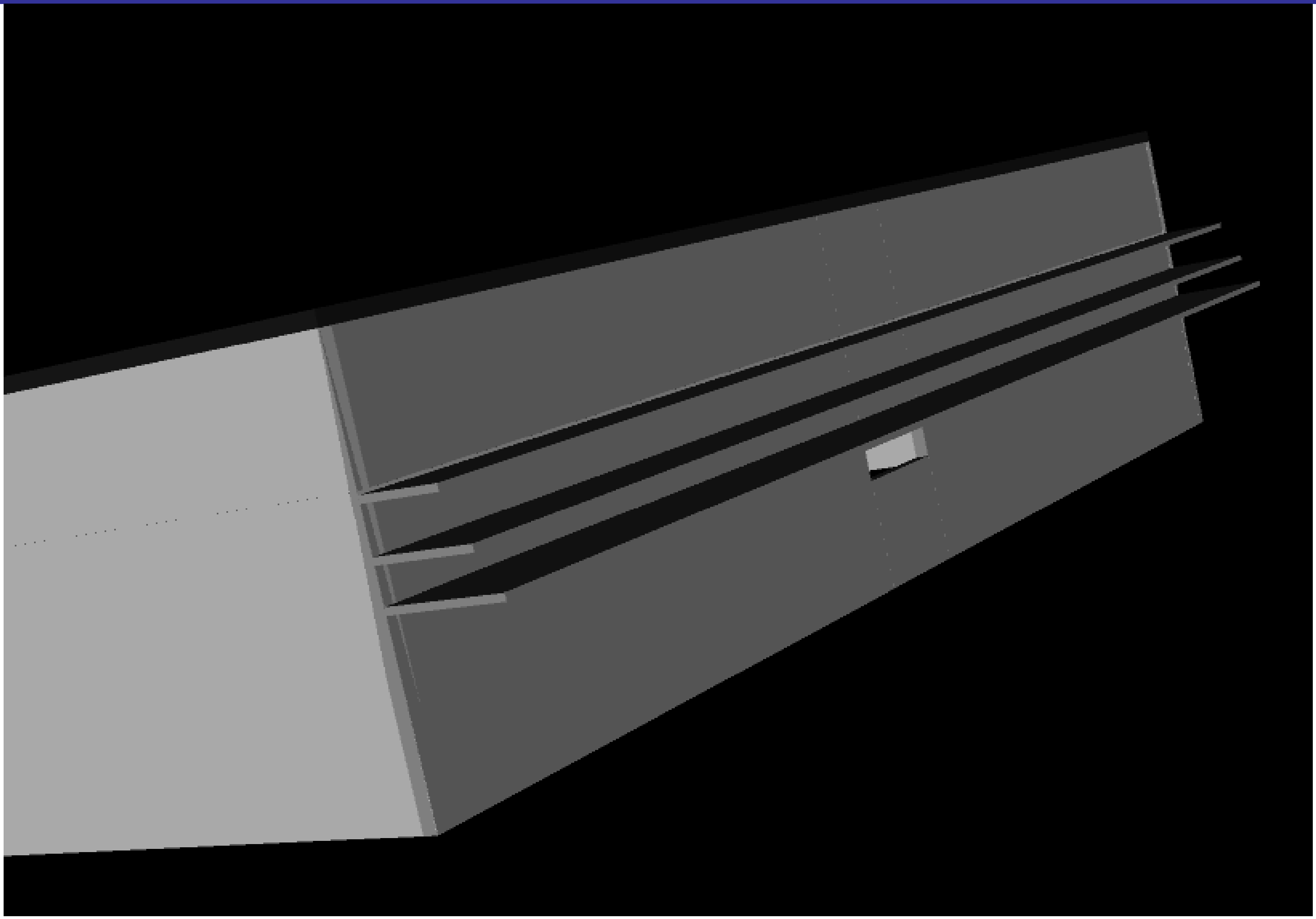
Example of design optimization



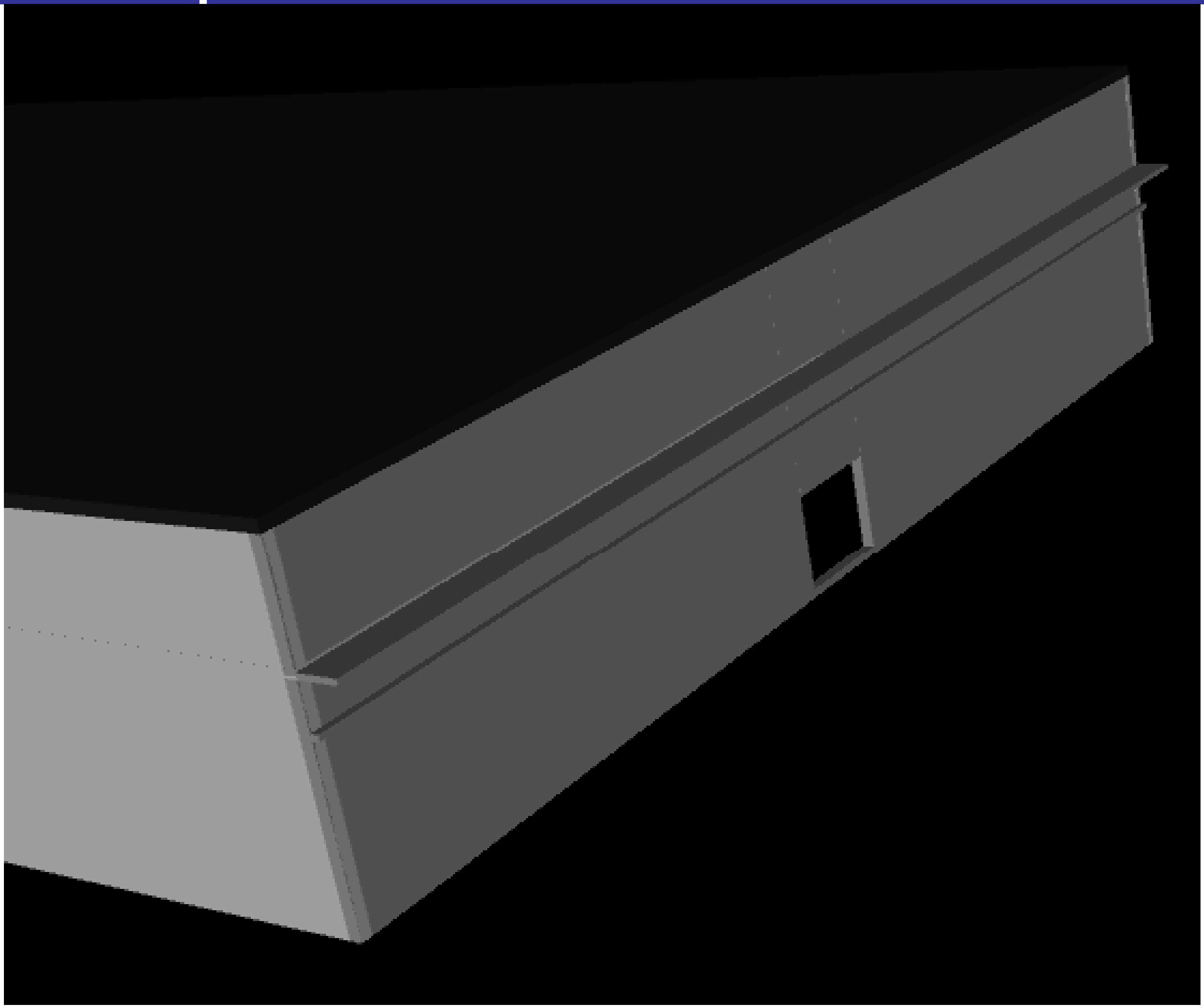
Optimal solution: lighting energy criterion



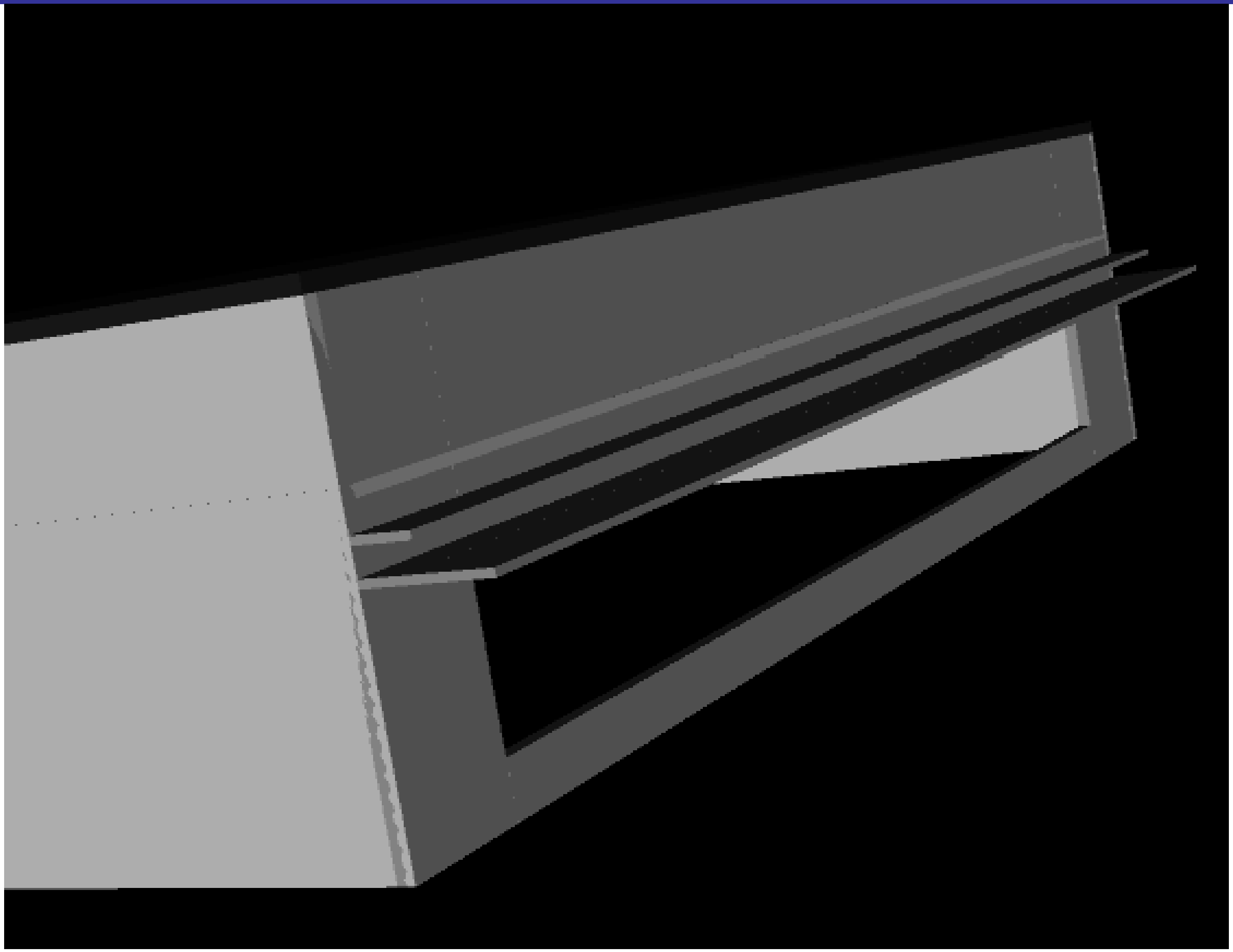
Optimal solution: solar thermal load criterion



Optimal solution: cost criterion



RR-PARETO3 solution



Conclusions

RR-PARETO3 algorithm is able to identify good compromise solutions that can be reasonably justified. It uses general principles and reasoning methods that are adopted by human decision makers in practice.

It can accommodate user preferences on the relative importance of objectives as well as acceptable deviations in objective function values.

It is a valuable tool for decision makers as well as for automatic processes