



upcomillas *es*

upcomillas *es*

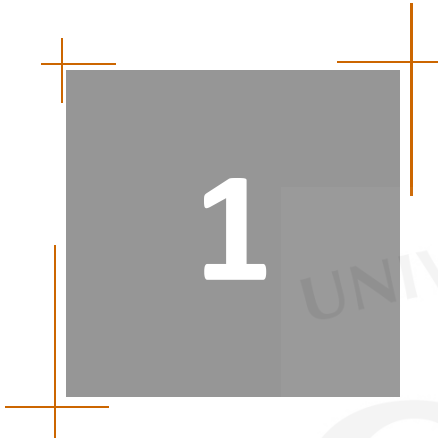
Multicriteria Decision Making

Andrés Ramos (Andres.Ramos@comillas.edu)
Pedro Sánchez (Pedro.Sanchez@comillas.edu)
Sonja Wogrin (Sonja.Wogrin@comillas.edu)

Contents

1. Basic concepts
2. Continuous methods
3. Discrete methods





Basic concepts

Continuous methods

Discrete methods



M A D R I D

Basic concepts



Decision theory

- Decision: choosing the best of possible → To state best and possible
- Possible solutions or feasible decisions:
 - Finite set: alternatives can be enumerated
 - Continuous set: alternatives are defined through constraints
- The Best:
 - One criterion (classic Optimization and classic Decision Theory)
 - Multiple criteria o multiple decision makers (Game Theory and Multicriteria Decision Making)

Which is the best animal of the nature in running, flying and swimming simultaneously?

- The fastest runner?



Cheetah is the fastest running animal in the world

- The fastest flying?



Peregrine falcon is the fastest bird in the world

- The fastest swimmer?



Sailfish is the fastest fish in the world

And the winner is ... the DUCK

- Is able to run although less than the cheetah
- Is able to fly although less than the peregrine falcon
- Is able to swim although less than the sailfish



Methods

- Continuous decisions
 - Multiobjective optimization
 - Compromise programming
 - Satisfying methods (goals)

- Discrete decisions
 - Analytical Hierarchic Process (AHP)
 - Outranking methods (Electre, Promethee, ...)

- General statement:
$$\text{opt } z = (z_1(x), \dots, z_p(x))$$
$$x \in F$$

- F : feasible decision space (continuous: feasible region, $F \subseteq \mathbb{R}^n$)
- $z(F)$: criterion space (numerical criteria: $z(F) \subseteq \mathbb{R}^p$)

Basic concepts

- **Attribute**: observable "value" (measurable) of an alternative, independent of the decision maker
- **Objective**: direction to improve an attribute (max. o min. if numerical; otherwise, **preferences**)
- **Target**: an acceptable level of achievement for an attribute
- **Goal**: combination of an attribute with its target
- **Criterion**: relevant attributes, objectives or goals to a decision problem

Solution concept

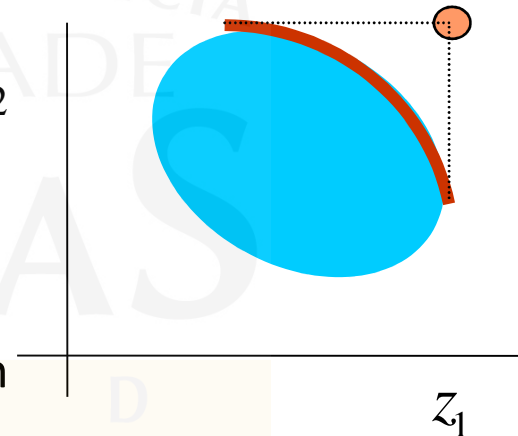
Efficiency or Pareto optimality criterion

One feasible solution is **efficient or Pareto optimal** if no other feasible solution can yield an improvement in one objective without causing a degradation in at least another objective

Dominated or non efficient alternative: there is another one with better attributes

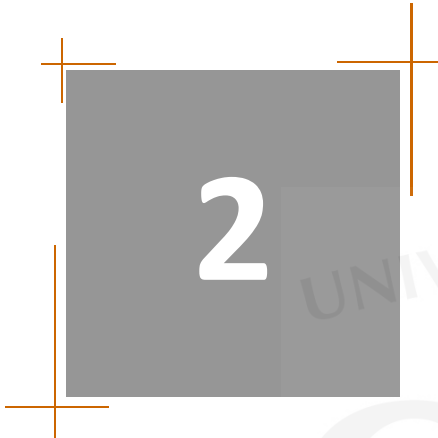
- Criteria space $\max (z_1, z_2)$
- Efficient set or Pareto frontier
- Dominated alternatives

Numerical attributes, objectives maximisation
→ **Efficient set:**



$$\mathcal{E} = \{x \in F : \nexists x' \in F \text{ with } z_k(x') \geq z_k(x) \forall k \text{ and } \exists t \in \{1, \dots, p\} \text{ with } z_t(x') > z_t(x)\}$$

Best compromise solution: efficient solution chosen by the decision maker



Basic concepts

Continuous methods

Discrete methods



Continuous methods



Multiobjective optimization

$$\max z = (z_1(x), \dots, z_p(x))$$

$$x \in F$$

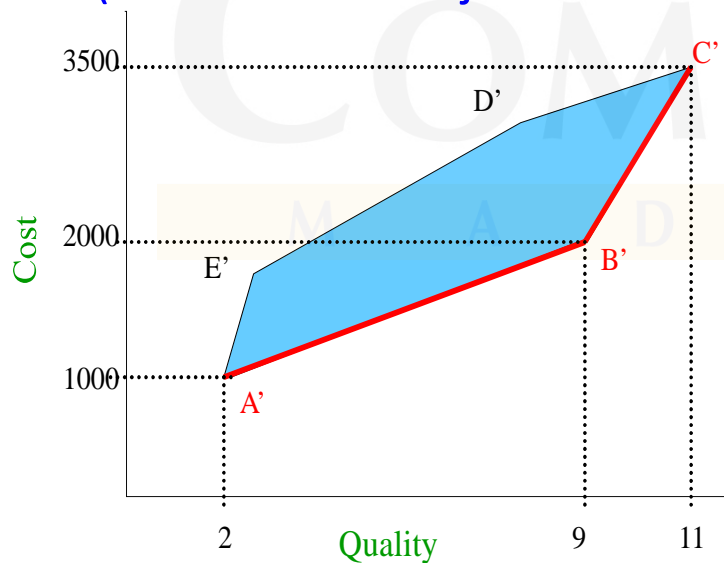
$z_i(x)$: mathematical expression for attribute i

x : decisional variables vector

F : constraints defining feasible solutions

Pay-off matrix: optimum value for an objective (without the others), and values of the other objectives for this solution. Conflict level

Trade-offs: degradation of one objective to improve another one in one unit ("cost" of one objective in terms of another one). Efficient set slopes



Pay-off matrix:

	Cost	Quality
Cost	1000	2
Quality	3500	11

Trade-offs:

$$T_{A'B'} = \frac{2000 - 1000}{9 - 2} = 142.86$$

$$T_{B'C'} = \frac{3500 - 2000}{11 - 9} = 750$$

Multiobjective optimization

Weighted-Sum Method (Zadeh, 1963)

Multiplying any objective function with a **weight** or non negative factor and **adding** in a single composite objective function

Parametric programming (changing weights the efficient set is obtained)

$$\left. \begin{array}{l} \max \sum_{i=1}^p \lambda_i z_i(x) \\ x \in F \\ \lambda \geq 0 \end{array} \right\} P(\lambda)$$

Theorem: If $\lambda_i > 0 \forall i$ then any optimal solution of $P(\lambda)$ is efficient.

Converse of theorem is true under some assumptions (convexity, linearity)

Normalized criteria (units)

Multiobjective optimization

Epsilon-Constraint Method

Optimisation of one objective including the rest of objectives as parametric constraints

Parametric programming (changing right hand side, efficient set)

$$\left. \begin{array}{l} \max_{x \in F} z_l(x) \\ z_k(x) \geq \varepsilon_k \quad k = 1, \dots, l-1, l+1, \dots, p \end{array} \right\} P_l(\varepsilon)$$

Theorem: If there is only one solution, it is efficient.

Theorem: If x^* is efficient, $\forall l \exists \varepsilon_k$ such that x^* is optimum of $P_l(\varepsilon)$

Multicriterion simplex method (Zeleny, 1974)

Only for linear objectives and constraints

Each iteration, check efficiency of solutions (extreme points or vertices)

All the efficient vertices are obtained

Efficient set: linear combination of adjacent vertices

Compromise programming

Ideal point: optimum values of each objective subject to problem constraints

$$z^* = (z_1^*, \dots, z_i^*, \dots, z_p^*) \quad z_i^* = \max_{x \in F} z_i(x) \quad z_i^* : \text{anchor value}$$

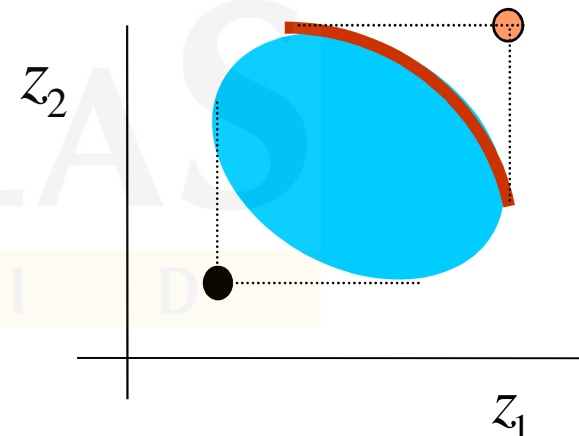
Optimum element or best-compromise solution: efficient solution closest to the ideal point (Zeleny's axiom of choice, 1976)

Degree of closeness between attribute i-th and its anchor value, normalized

z_{*i} anti-ideal or nadir point (worst value criteria in efficient set)

$$d_i(x) = \frac{z_i^* - z_i(x)}{z_i^* - z_{*i}}$$

$$\min_{x \in F} L_\pi = \left[\sum_{i=1}^p w_i \pi \left(\frac{z_i^* - z_i(x)}{z_i^* - z_{*i}} \right)^\pi \right]^{1/\pi}$$



Weights, importance of discrepancy of criteria (subjective → ordering, Saaty, ...)

$\pi = \infty$ → Tchebychev or minimising maximum distance (linear)

$\pi = 1$ → lineal addition of weighted distances (linear)

Compromise set: varying π (usually $[L_1, L_\infty]$)

Goal programming

Satisfying logic (Simon, 1955):

decision-making behaviour where, instead of attempting to optimize system performance, a level of aspiration is set either subjectively or heuristically and no further effort is expended to exceed that level of performance → **Goal programming** (Charnes y Cooper(61), Lee (72) e Ignizio (76))

Attribute → mathematical expression: $z_i(x)$

Target or level of aspiration: acceptable level of achievement \hat{z}_i

Goal: $z_i(x) \geq \hat{z}_i$

With **deviation variables**: $z_i(x) + n_i - p_i = \hat{z}_i$

Deviation variables to be minimized:

if goal is "at least" one value, n_i ; if it is "at most", p_i

Model (if at least):

$$\min_{x \in F \cap \text{goal constraints}} \sum_{i=1}^p n_i$$

Andrés Ramos

<http://www.iit.upcomillas.es/aramos/>

Andres.Ramos@comillas.edu

Pedro Sánchez

Pedro.Sanchez@comillas.edu

Sonja Wogrin

Sonja.Wogrin@comillas.edu

Departamento de Organización Industrial

Alberto Aguilera, 23
28015 Madrid, Spain
Tel +34 91 542 28 00
Fax + 34 91 542 11 32
info-doi@doi.icaei.upcomillas.es

www.upcomillas.es

