What is a Decision Problem?

Alberto Colorni[1] Alexis Tsoukiàs[2]

[1]INDACO, Politecnico di Milano,
alberto.colorni@polimi.it

[2]LAMSADE - CNRS, Université Paris-Dauphine
tsoukias@lamsade.dauphine.fr

Napoli, 28/04/2013

Outline

Problems

- Patients triage in emergency room;
- Identification of classes of similar DNA sequences;
- Star ratings of hotels;
- Waste collection vehicle routing;
- Vendor rating and bids assessment;
- Optimal mix of sausages;
- Chip-set lay out;
- Airplanes priority landing;
- Tennis tournament scheduling ...

What is a decision problem?

Consider a set A established as any among the following:

- an enumeration of objects;
- a set of combinations of binary variables (possibly the whole space of combinations);
- a set of profiles within a multi-attribute space (possibly the whole space);
- a vector space in $\mathbb{R}^n$.

Technically:

A Decision Problem is a partitioning of A under some desired properties.

What is important?

What does really matter?

In designing, choosing, applying, implementing, understanding, explaining, justifying, a method?

What are the primitives?

And what is the derived information and the expected outcomes?

What is important?

What does really matter?

In designing, choosing, applying, implementing, understanding, explaining, justifying, a method?

What are the primitives?

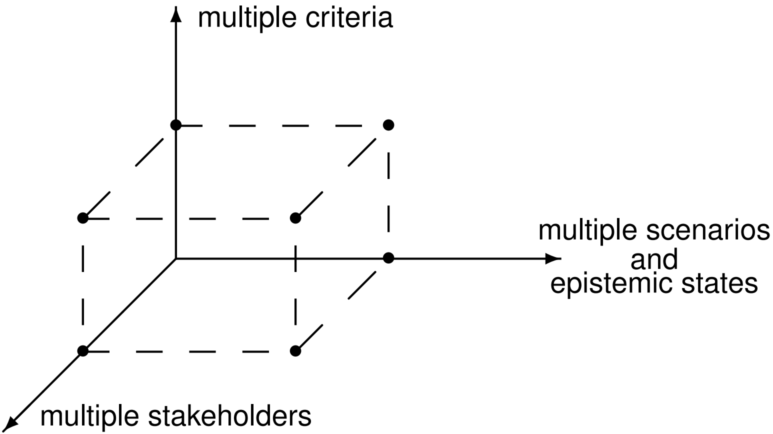
And what is the derived information and the expected outcomes?

Why is not straightforward?

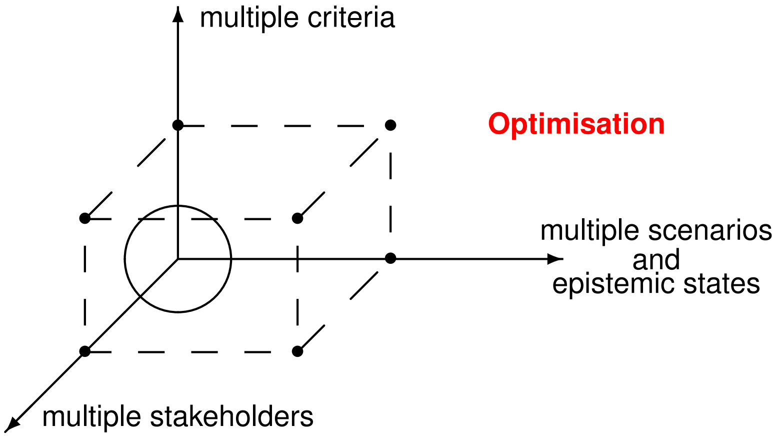
- multiple opinions
- multiple values
- multiple likelihoods

+ algorithmic aspects

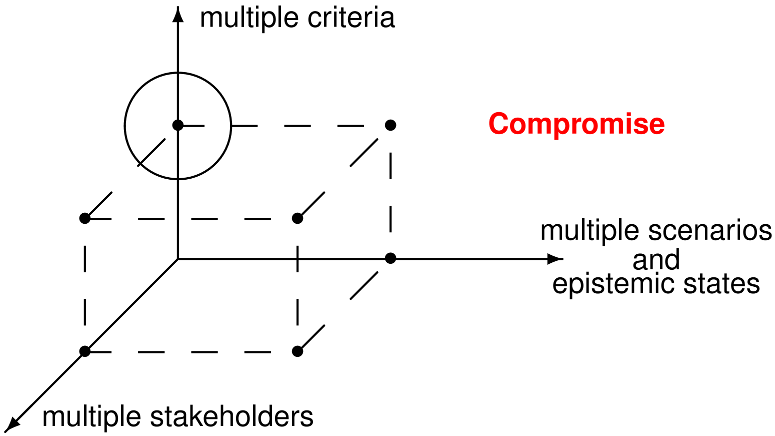
Partitioning? How?



Partitioning? How?

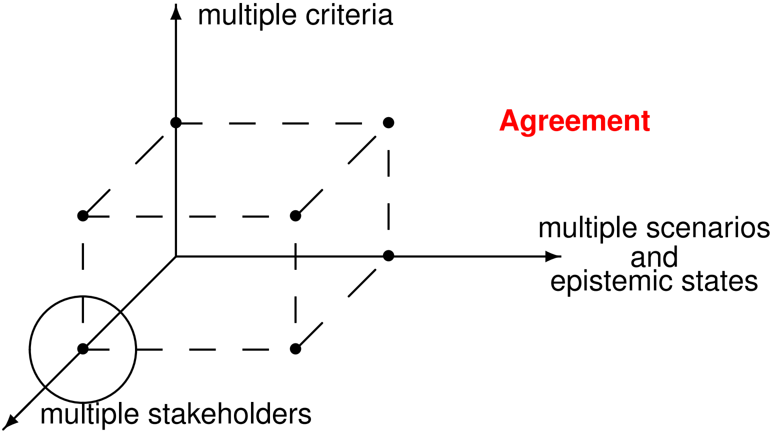


Partitioning? How?



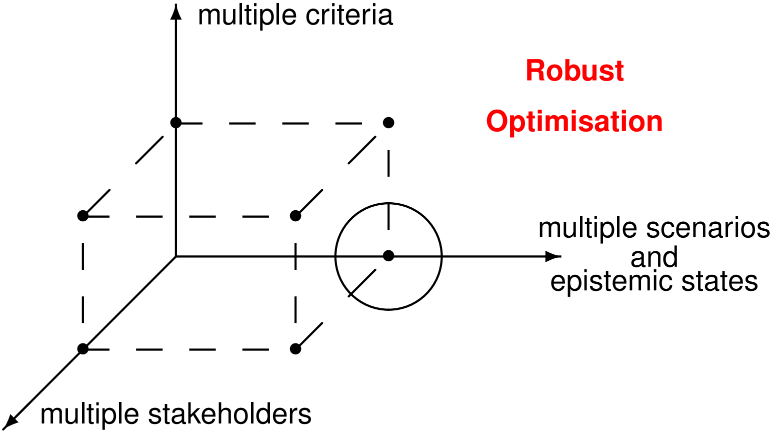
Compromise

Partitioning? How?



Agreement

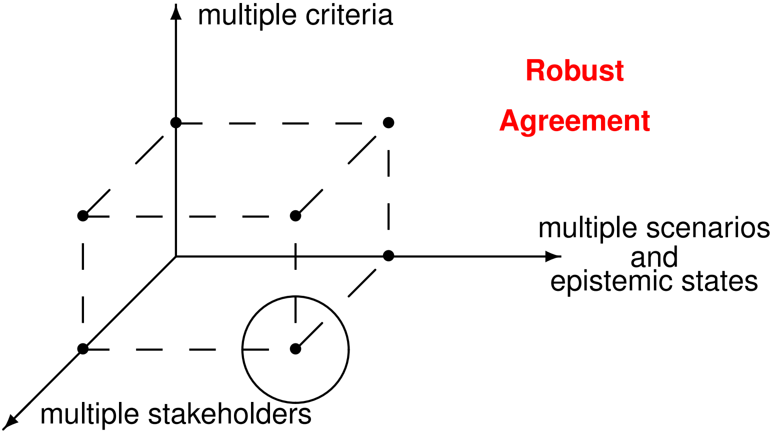
Partitioning? How?



Robust

Optimisation

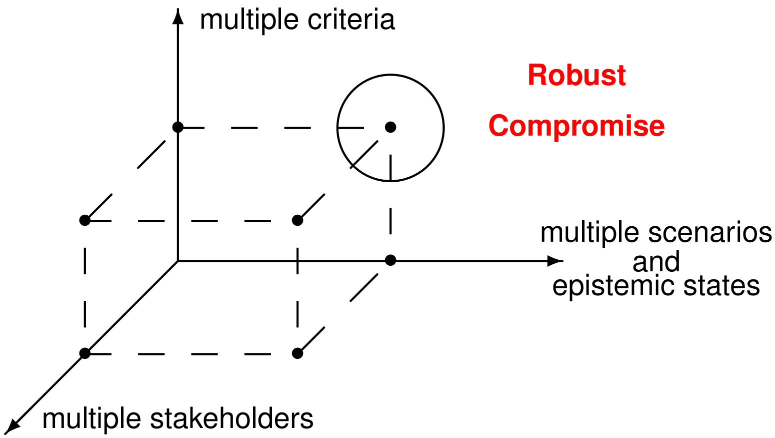
Partitioning? How?



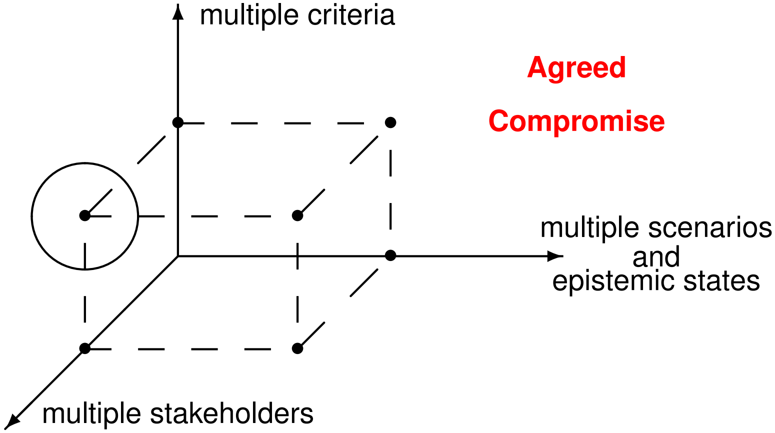
Robust

Agreement

Partitioning? How?



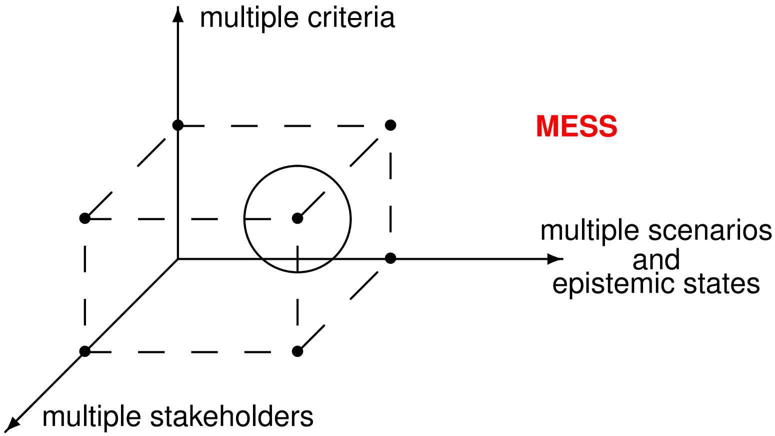
Partitioning? How?



Agreed

Compromise

Partitioning? How?



Is that all?

- Behind a criterion other criteria may be considered in a hierarchy of criteria (objectives);
- Behind a stakeholder other actors may have to be considered, that precise stakeholder being a speaker for a community;
- Behind a state of the nature other uncertainties may have to be considered;
- Any combination of the above may in reality occur as complex as possible.

Claims

Claim 1

All the previously mentioned problems boil down in aggregating some ordering relations applied on the set A.

Claim 2

Establishing the set A is on its turn a decision problem. We explore one step of the recursion without any loss of generality.

Claim 3

From an algorithmic point of view a decision problem boils down to an optimisation algorithm.

Claims

Claim 1

All the previously mentioned problems boil down in aggregating some ordering relations applied on the set A.

Claim 2

Establishing the set A is on its turn a decision problem. We explore one step of the recursion without any loss of generality.

Claim 3

From an algorithmic point of view a decision problem boils down to an optimisation algorithm.

Claims

Claim 1

All the previously mentioned problems boil down in aggregating some ordering relations applied on the set A.

Claim 2

Establishing the set A is on its turn a decision problem. We explore one step of the recursion without any loss of generality.

Claim 3

From an algorithmic point of view a decision problem boils down to an optimisation algorithm.

Primitives

- The set A.
- The description of the elements of A.
- Preference (ordering) statements about A and its subsets.
- Preference statements are of two types: relative and absolute ones.

Critical Issues

- The set of alternatives
- Problem statement
- Differences of preferences
- Hierarchy/Separability/Indipendence
- Positive and Negative Reasons

Partitioning? For what?

Practically we partition A in n classes. These can be:

	Pre-defined wrt some external norm	Defined only through pairwise comparison
Ordered	Rating	Ranking
Not Ordered	Assigning	Clustering

Two special cases:
- there are only two classes (thus complementary);
- the size (cardinality) of the classes is also predefined.

What is a ranking problem?

Primitive

The primitive is a binary relation on A: $\succeq \subseteq A \times A$ to be read "at least as good as".

Result

The result is a partitioning of A in $[A_1], \cdots [A_n]$ such that:
$[A_j] \geq [A_i] \Leftrightarrow j \geq i$ and
$\forall x \in [A_j], y \in [A_i] : x \succeq' y$

Discussion 1

What is a choice problem?

We partition A in two classes $[A_1] \geq [A_2]$. Thus $[A_1] = \sup_A(\succeq')$.

What is an optimisation problem?

A choice problem for which:
- $\succeq = \succeq'$
- $x \succeq y \iff f(x) \geq f(y)$.
- Thus $[A_1] = \max_A f(x)$

Discussion 1

What is a choice problem?

We partition A in two classes $[A_1] \geq [A_2]$. Thus $[A_1] = \sup_A(\succeq')$.

What is an optimisation problem?

A choice problem for which:

- $\succeq = \succeq'$
- $x \succeq y \Leftrightarrow f(x) \geq f(y)$.
- Thus $[A_1] = \max_A f(x)$

Discussion 2

Why is $\succeq'$ different from $\succeq$?

Generally speaking $\succeq$ is not an ordering relation since preferences can be partial and or inconsistent. If we have to proceed with some operational procedure we need to transform $\succeq$ to an ordering relation $\succeq'$.

Discussion 2

Why is $\succeq'$ different from $\succeq$?

Generally speaking $\succeq$ is not an ordering relation since preferences can be partial and or inconsistent. If we have to proceed with some operational procedure we need to transform $\succeq$ to an ordering relation $\succeq'$.

How do we learn $\succeq$?

Discussion 2

Why is $\succeq'$ different from $\succeq$?

Generally speaking $\succeq$ is not an ordering relation since preferences can be partial and or inconsistent. If we have to proceed with some operational procedure we need to transform $\succeq$ to an ordering relation $\succeq'$.

How do we learn $\succeq$?
What properties should $\succeq'$ fulfill?

What is a clustering problem?

Primitive

The primitive is a set of binary relations on A: $\approx_l \subseteq A \times A$ to be read “similar to”.

Result

The result is a partitioning of A in $[A_1], \cdots [A_n]$ such that:
$\exists \approx_l : \forall x, y \in [A_j] \;\; x \approx y$ and
$\forall x \in [A_j],\; y \in [A_i] : \neg(x \approx y)$

Discussion 1

Indiscernibility.

In case $\approx_l$ are equivalence relations then the partitioning of A results in constructing the indiscernibility relation on A. However, this is not generally the case and $[A_j] = \sup_A(\approx_l)$.

In other terms we try to maximise similarity within classes (clusters) and minimise similarity among classes (clusters).

Discussion 1

Indiscernibility.

In case $\approx_I$ are equivalence relations then the partitioning of A results in constructing the indiscernibility relation on A. However, this is not generally the case and $[A_j] = \sup_A(\approx_I)$.

In other terms we try to maximise similarity within classes (clusters) and minimise similarity among classes (clusters).

Discussion 2

Distances.

If $\approx_l$ are nested similarity relations with nice properties then we can establish a metric:
- $s(x, y)$: how similar is x to y?
- $d(x, y)$: how distant is x from y?
Then $[A_y] = \{x | \max_A F(s(x, y))\}$,
F being a measure of the overall similarity of the elements of $[A_y]$ with respect to y.

What properties should F and the metrics fulfill?

What is a rating problem?

Primitive

The primitive is a binary relation on A: $\succeq \subseteq A \times P \cup P \times A$
to be read "at least as good as".
P being the set of external "norms" characterising the ordered classes $C_1 \rhd \cdots \rhd C_n$

Result

The result is to assign each element of A in a C_j such that:
$x \in C_j \;\Leftrightarrow\; x \succeq' p_j, p_{j+1}, \cdots p_n$ and $p_1 \cdots p_{j-1} \succeq' x$

Discussion 1

Constraint Satisfaction

If $\forall x, y \in A \cup P \;\; x \succeq y \;\Leftrightarrow\; f(x) \geq f(y)$.
Then $x \in C_j \;\Leftrightarrow\; f(p_{j-1}) \geq f(x) \geq f(p_j)$.
This is a Constraint Satisfaction Problem.

Why is $\succeq'$ different from $\succeq$?

Generally speaking $\succeq$ is not an ordering relation since preferences can be partial and or inconsistent. If we have to proceed with some operational procedure we need to transform $\succeq$ to an ordering relation $\succeq'$.

Discussion 1

Constraint Satisfaction

If $\forall x, y \in A \cup P \;\; x \succeq y \;\; \Leftrightarrow \;\; f(x) \geq f(y)$.
Then $x \in C_j \;\; \Leftrightarrow \;\; f(p_{j-1}) \geq f(x) \geq f(p_j)$.
This is a Constraint Satisfaction Problem.

Why is $\succeq'$ different from $\succeq$?

Generally speaking $\succeq$ is not an ordering relation since preferences can be partial and or inconsistent. If we have to proceed with some operational procedure we need to transform $\succeq$ to an ordering relation $\succeq'$.

What is an assigning problem?

Primitive

The primitive is a set of binary relations on A: $\approx_l \subseteq A \times P \cup P \times A$ to be read "similar to".
P being the set of external "norms" characterising the classes $C_1 \cdots C_n$

Result

The result is to assign each element of A in a C_j such that:
$x \in C_j \;\Leftrightarrow\; \exists \approx_l : x \approx_l p_j$

Discussion 1

Constraint Satisfaction

If $\forall x, y \in A \cup P \;\; x \approx_I y \;\Leftrightarrow\; f(x) = f(y)$.
This is once again a Constraint Satisfaction Problem.

Basic Claim

- **Any unsupervised decision problem is an optimisation problem.**
- **Any supervised decision problem is a constraint satisfaction problem.**

Since any constraint satisfaction problem can be seen as an optimisation problem, we can definitely focus only to the later ones

When in reality we just know that:

$$
\begin{array}{rcccl}
w\ {}_1 z & {}_1 & x & {}_1 y & {}_1 t \\
w & {}_2 & y & {}_2 x & {}_2 t \ {}_2 z \\
w\ {}_3 t & {}_3 & x & {}_3 y & {}_3 z \\
z & {}_4 & y & {}_4 x & {}_4 t \ {}_4 w \\
 & & \vdots & &
\end{array}
$$

The Problem

Suppose we have n ordering relations $\succeq_1 \cdots \succeq_n$ on the set A. We are looking for an overall ordering relation $\succeq$ on A "representing" the different orders.

$$\begin{tikzcd}
\succeq_i (x,y) \arrow[r, leftrightarrow] \arrow[d, leftrightarrow] \arrow[dr, leftrightarrow] & f_i(x), f_i(y) \arrow[d, leftrightarrow] \\
\succeq (x,y) \arrow[r, leftrightarrow] & F(x,y)
\end{tikzcd}$$

Two fundamental questions

1. **How do we consider preferences and differences of preferences along a single criterion/dimension?**
2. **How do we consider preferences and differences of preferences among several different criteria/dimensions?**

Two fundamental questions

1. **How do we consider preferences and differences of preferences along a single criterion/dimension?**
2. **How do we consider preferences and differences of preferences among several different criteria/dimensions?**