Decision Deck

Multiple Criteria Decision Aiding in action

Decision Deck & diviz

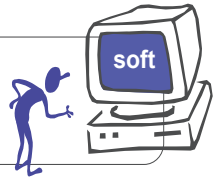
Miłosz Kadziński

milosz.kadzinski@cs.put.poznan.pl

1 : Institute of Computing Science, Poznan University of Technology, Poznań, Poland
2 : Decision Deck Consortium
3 : EURO Working Group on MCDA & International MCDM Society
4 : Participant of MCDA/M Summer School in Paris (2010)

Agenda

How does software situation look in MCDA?
- many methods / software
- great need for unified software framework



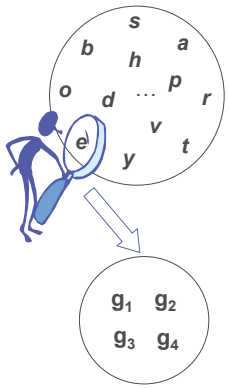
Decision Deck – *Decision what*? ? ? ?
- XMCDA, MCDA web services, diviz
- ***diviz***: design, execution and deployment tool
- live demo and "**hands on training**"

A bit of methodological summary for illustrative purpose
- focus on value- and outranking based methods
- "reinvent" methods on your own

www.cs.put.poznan.pl/mkadzinski/MCDASummerSchool/

Essence of MCDA

***A* = set of alternatives**



***G* = set of criteria**

INPUT

Alternatives are evaluated on multiple preference dimensions (**criteria**, attributes)

OUTPUT

Help to work out the **recommendation**, i.e. to determine the best alternatives, rank them or assign to ordered **classes**

INPUT

By taking into account the **preferences** of the decision maker

sorting

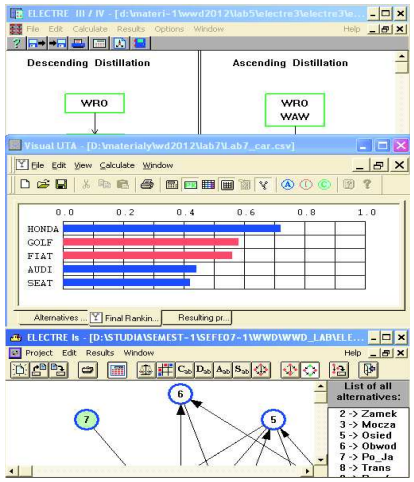
w u a	C_3=leading
v y d o e	C_2=average
h b t p	C_1=weak

ranking

w u
a
v y
d o
e
h b
t p

Software situation in MCDA

- many different methods
- many **separate** software products
  - heteregeneous user interfaces
- no standard data format and unified software to test the same problem on various methods
- many algorithms **not easily available**
- often **not free** (financial and open source)
- existing MCDA methods cannot communicate



MCDA software overview

UTA UTA+, Visual UTA, Right Choice, DECERNS, UTADIS	**ELECTRE** Electre Is, Electre III-IV, Electre Tri, IRIS, MCDA-ULaval	JSMAA VIP (MAVT) M-MACBETH jMAF (DRSA) 1000 minds Quantum-GIS plugins …
AHP/ANP Make It Rational, Web HIPRE, Expert Choice, Decision Lens, Super Decision	**PROMETHEE** Decision Lab, D-Sight, Smart Picker Pro, Visual Promethee, DECERNS	

Check software sections at the websites of EWG-MCDA and MCDM society:

- http://www.cs.put.poznan.pl/ewgmcda/
- http://www.mcdmsociety.org/

A. Ishizaka, P. Nemery, *Multi-criteria Decision Analysis: Methods and Software*, Wiley, 2013

Decision Deck project

Decision Deck project
aims at collaboratively developing open source software tools implementing Multiple Criteria Decision Aiding methods and concepts

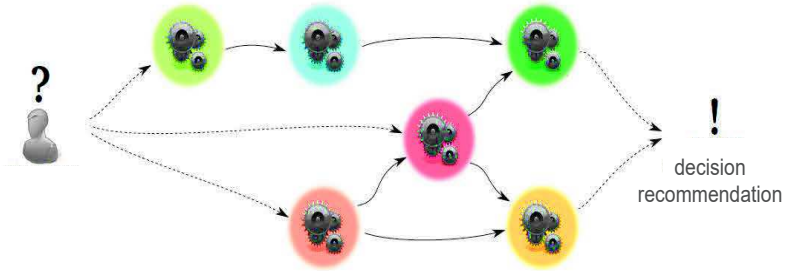
Its **purpose** is to provide effective tools to three types of users:

- **practitioners** (consultants / analysts) who use MCDA tools to support actual decision makers
- **researchers** who want to test, compare or develop methods
- **teachers and students** who present / use MCDA methods in courses

Promote MCDA research and make it more visible to the "outside" world

How are MCDA methods designed?

- MCDA methods are **sequences** of *elementary* algorithms
- MCDA methods *share* a lot of **similarities**
- MCDA methods need to be **adaptable** to the given practical case



Focus on three initiatives from Decision Deck project

XMCDA – to make algorithms interoperable
- a data standard for MCDA
- standardized format to represent objects and data structures issued from MCDA

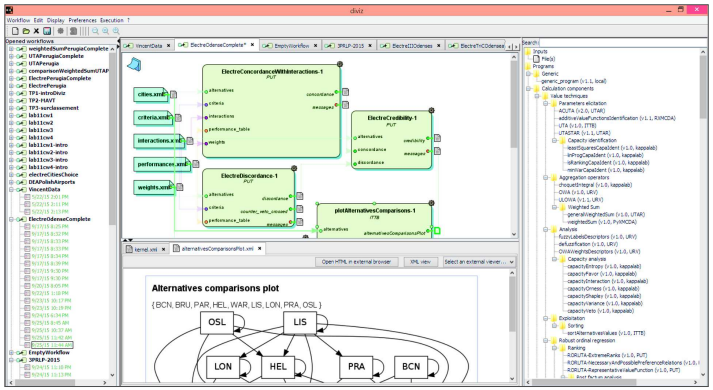
XMCDA web services –
– to make algorithms easily available
- algorithmic components or complete MCDA methods accessible online
- reuse of existing implementations

diviz – to create complex algorithmic workflows
- open source Java client and server
- web services compositions, workflow management and deployment

diviz software

Just download it and run - www.diviz.org/download

- no need to install (although possible, for different operating systems)
- **platform independent *jar***
- requirement for the **Internet access** and Java installed (=commonplace)



Lecture aims

diviz

- open source Java client and server
- a tool for designing complex MCDA workflows via the XMCDA web-services

- study a *classical* multiple criteria decision problem: **Thierry's car choice problem**
- **learn** how to use the **diviz software**
- **use** diviz as a decision support tool

The data = Thierry's choice problem

- In 1993, **Thierry**, a student aged 21, is passionate about sports cars and **wishes to buy** a middle range 4 years old car with a powerful engine
- He selects **three viewpoints** related to cost (criterion g_1), performance of the engine (criteria g_2 and g_3) and safety (criteria g_4 and g_5)
- The cost criterion g_1 (€) and the performance criteria acceleration g_2 (seconds) and pick up g_3 (seconds) have **to be minimized**, whereas the safety criteria brakes g_4 and road-hold g_5 have **to be maximized**
- The values of the safety criteria are average evaluations obtained from multiple qualitative evaluations which have been re-coded as integers between 0 and 4

D. Bouyssou, T. Marchant, M. Pirlot, P. Perny, A Tsoukias, P. Vincke, *Evaluation and Decision Model, A critical perspective*, Kluwer, 2000

Thierry's choice - performance matrix

			five criteria				
	car ID	car name	cost ($g1$, €)	accel. ($g2$, s)	pick up ($g3$, s)	brakes ($g4$)	road-hold ($g5$)
fourteen alternatives	a01	Tipo	18342	30.7	37.2	2.33	3
	a02	Alfa	15335	30.2	41.6	2	2.5
	a03	Sunny	16973	29	34.9	2.66	2.5
	a04	Mazda	15460	30.4	35.8	1.66	1.5
	a05	Colt	15131	29.7	35.6	1.66	1.75
	a06	Corolla	13841	30.8	36.5	1.33	2
	a07	Civic	18971	28	35.6	2.33	2
	a08	Astra	18319	28.9	35.3	1.66	2
	a09	Escort	19800	29.4	34.7	2	1.75
	a10	R19	16966	30	37.7	2.33	3.25
	a11	P309-16	17537	28.3	34.8	2.33	2.75
	a12	P309	15980	29.6	35.3	2.33	2.75
	a13	Galant	17219	30.2	36.9	1.66	1.25
	a14	R21t	21334	28.9	36.7	2	2.25

Table: **Which car should Thierry buy?**

Multi-Atttribute Value Theory

Natural **extension of the weighted sum** which takes into acount the non-linearity of preferences:

$aPb \leftrightarrow U(a) > U(b)$

$a I b \leftrightarrow U(a) = U(b)$

where $U(a) = f(u_1(g_1(a)), \ldots, u_n(g_n(a)))$

Various possible aggregation models, but here:

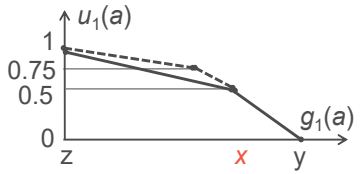
$$U(a) = \sum_j w_j \cdot u_j(g_j(a)) = w_1 \cdot u_1(g_1(a)) + \ldots + w_n \cdot u_n(g_n(a))$$

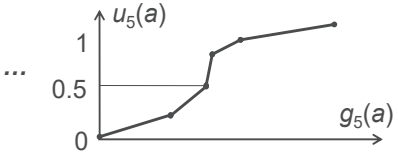
- w_j: weight associated with criterion g_j
- u_j: marginal value function associated with g_j
- $g_j(a)$: performance of alternative a on g_j

MAVT - marginal (partial) value functions

Step 1:

- **Determine a value function u_j** for each criterion such that $u_j(g_j(a))$ represents the value of a on criterion g_j, and **read off $u_j(g_j(a))$** for $g_j(a)$
- The **u_j repesents the decision maker's preferences** (and not a normalization of the data)





For example: **bisection method**

- Define the performances that correspond to **values 0 and 1**
- Indicate a performance x such that changing from the 0-value performance to x increases the value as much as changing from x to the 1-value performance - the **selected midpoint** corresponds to value **0.5**
- Use **the same process** to bisect the interval of [0,0.5] and/or [0.5,1], etc.

MAVT - weights

Step 2: determine the weights (scale coefficients) w_j

For example: **Rank Order Centroid (ROC) method**

- Order the criteria from **the most to the least important**

$$w_1 > w_2 > w_3 > w_4 > w_5$$

- Compute the weight for criterion with rank r_k as follows:

$$w(r_k) = 1/n \sum_{j=k\ldots n} 1/j$$

$$w(r_1) = 1/5 \sum_{j=1\ldots5} 1/j = 1/5\ (1/1 + 1/2 + 1/3 + 1/4 + 1/5) = 0.457$$

$$w(r_2) = 1/5 \sum_{j=2\ldots5} 1/j = 1/5\ (1/2 + 1/3 + 1/4 + 1/5) = 0.257$$

$w(r_3) = 0.157$, $w(r_4) = 0.09$, $w(r_5) = 0.04$

- Weights reflect the **centroid** (centre of mass**) of the simplex defined by the ranking of the criteria**; they are normalized to sum up to 1

Step 3: compute the comprehensive value of each alternative

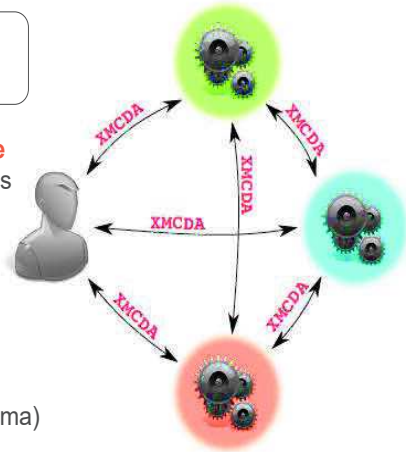
$$U(a) = \sum_j w_j \cdot u_j(g_j(a)) = w_1 \cdot u_1(g_1(a)) + \ldots + w_n \cdot u_n(g_n(a))$$

Goals of XMCDA standard

XMCDA

- **a data standard for MCDA**

- A unique communication **language** with and between MCDA algorithms
- **Standarization** and **unification** of multiple schools of thought
- Representation of MCDA data elements in XML according to a grammar (the XMCDA XML schema)



How to define MCDA inputs in XMCDA?

EXEMPLARY INPUTS

- **MCDA concept = the list of alternatives**

```
<alternatives>
   <alternative id="a01" name="TIPO">
      <type> real </type>
   </alternative>
   ...
   <alternative id=„fictiveBest" name="IDEAL ALTERNATIVE">
      <type> fictive </type>
   </alternative>
</alternatives>
```

XMCDA types = structures created to represent MCDA concepts

- **MCDA concept = criteria weights**

```
<criteriaValues mcdaConcept="Importance" name="significance">
   <criterionValue>
      <criterionID> g1 </criterionID>
      <value>
         <real> 0.457 </real>
      </value>
   </criterionValue>
   ...
</criteriaValues>
```


How are MCDA outputs represented in XMCDA?

EXEMPLARY OUTPUTS

- **MCDA concept = ranks of the alternatives**

```
<alternativesValues mcdaConcept="alternativesRanks">
   <alternativeValue>
     <alternativeID> a01 </alternativeID>
     <value>
       <real> 3 </real>
     </value>
   </alternativeValue>
   ...
</alternativesValues>
```

XMCDA types = structures created to represent MCDA concepts

- **MCDA concept = pair-wise (preference, outranking) relations**

```
<alternativesComparisons>
   <pairs>
     <pair>
       <initial> <alternativeID> a01 </alternativeID> </initial>
       <terminal> <alternativeID> a02 </alternativeID> </terminal>
     </pair>
     ...
   </pairs>
</alternativesComparisons>
```


Quick guide to XMCDA

- Possible to store advanced preference information on alternatives, criteria, and classes as well results typical for MCDA applications
- For details, see **http://www.decision-deck.org/xmcda**
- In particular, have a look at the ***Quick guide to XMCDA***
- Work with **examples available on-line** (whenever anyone is using XMCDA, (s)he is obliged to make the examples available for testing purposes)
- In order to avoid the writing of XMCDA, **csvToXMCDA-* converters** are available (see practical work hereafter)

diviz demo (1)

Help Thierry to **choose the car** which is ”best” for him

TIME FOR DEMO

MAVT

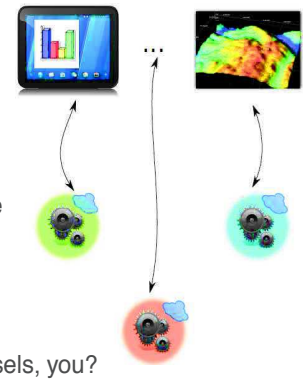
Motivation for XMCDA web services

XMCDA web-services

- MCDA algorithms which are made available for anybody over the Internet
- Reuse of existing implementations

MCDA researchers

- are often not computer scientists
- have programmed their algorithms in the programming language they know best



Idea

- allow researchers publishing their programs online
- require input / output in the XMCDA format

Maintained by the IMT Atlantique diviz team

Contributors: Poznań, Brest, Paris, Luxembourg, Tarragona, Mons, Rotterdam, Lyon, Coimbra, Brussels, you?

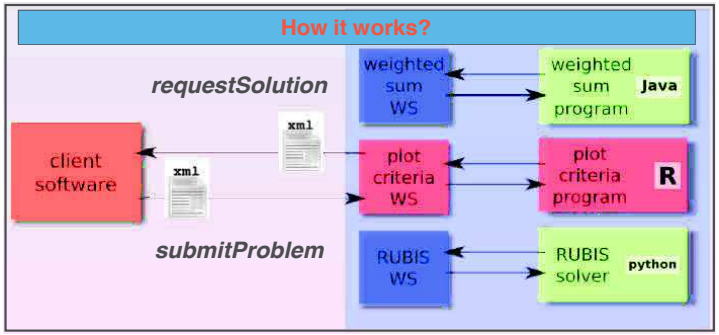
How to use XMCDA web services?

How to use XMCDA web services?

- Via various client softwares, in particular via *diviz*

What data is exchanged?

- XML files respecting the XMCDA standard



What are the main advantages?

- Heavy calculations on a distant server in France
- Output of a web service can be reinjected into another web service

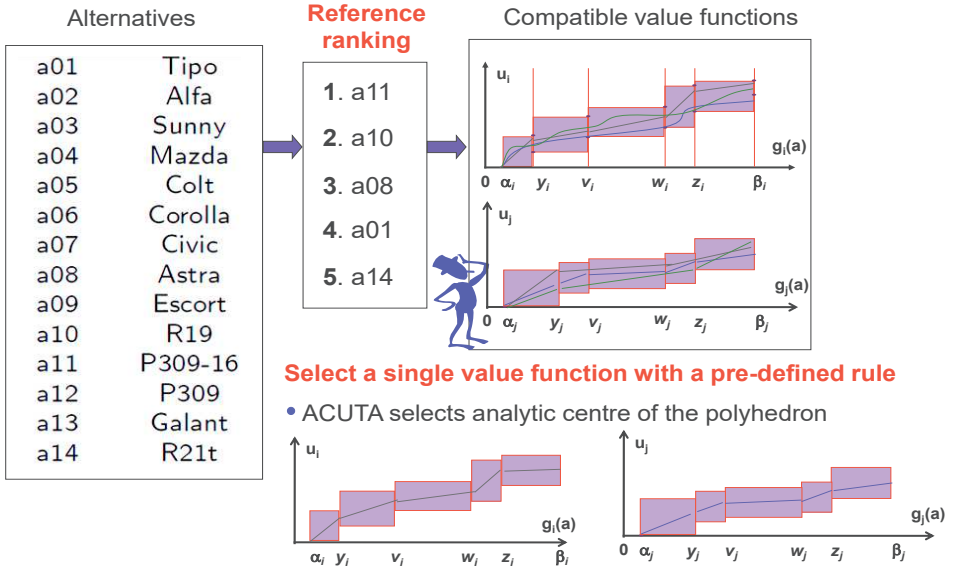
Why XMCDA web services are useful?

MCDA web services

- MCDA algorithms which are made available for anybody over the Internet
- reuse of existing implementations

- **Elementary procedures/algorithms available as separate software pieces**
- **If properly chained, they would rebuild the original method**
  - Remove the black box effect of certain software
  - Better understand the heart of the methods
  - Avoid repeated implementation of the same algorithms

UTA-like methods



Select a single value function with a pre-defined rule

- ACUTA selects analytic centre of the polyhedron

UTA-like methods: step by step

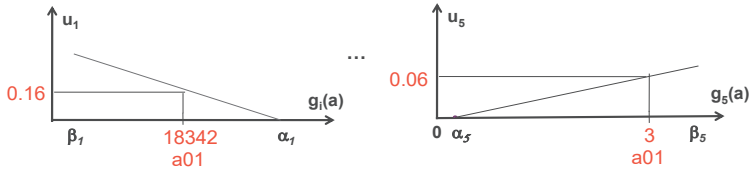
Step 1: **Provide preference information:**
ranking (pairwise comparisons) **of reference alternatives**
(e.g., a11 > a10 > a08 > a01 > a14)
and **number of segments** for each marginal value function
(e.g., all marginal functions are linear = 1 linear piece)

Step 2: **Select a central value function** according to a pre-defined rule
for example, **ACUTA** selects an analytic centre (UTAMP, UTASTAR, …)

Step 3: Compute **marginal values** for all alternatives
for example, $u_1(a01) = u_1(18323) = 0.16$, etc.

Step 4: Compute **comprehensive values**
for example, $U(a01) = u_1(a01) + \ldots + u_5(a01) = 0.16 + \ldots + 0.06 = 0.45$

Step 5: **Rank alternatives** w.r.t. their comprehensive values
for example, **1. a03 - 0.73, 2. a11 - 0.71, …, 12. a01 – 0.45, etc.**



More advanced preference information in XMCDA

XMCDA
- modeling reference ranking
  - a11> a10 > a08 > ...
- defining shape of marginal value functions
  - one segment for g_1 and g_2

```
<alternativesValues>
...
<alternativeValue>
<alternativeID>a11</alternativeID>
   <value>
      <integer>1</integer>
   </value>
</alternativeValue>
<alternativeValue>
<alternativeID>a10</alternativeID>
   <value>
      <integer>2</integer>
   </value>
</alternativeValue>
...
</alternativesValues>
```

```
<criteriaValues
     mcdaConcept="numberOfSegments">
      <criterionValue>
    <criterionID>g1</criterionID>
         <value>
         <integer>1</integer>
       </value>
      </criterionValue>
   <criterionValue>
     <criterionID>g2</criterionID>
         <value>
           <integer>1</integer>
         </value>
    </criterionValue>
   ...
<criteriaValues>
```


diviz demo (2)

TIME FOR DEMO
UTA & ACUTA

Essence of diviz software

diviz
- open source Java client and server
- web services compositions, workflow management and deployment

Available components = algorithmic elements available via XMCDA web services

- Calculation components, e.g. aggregation operators, post-analysis elements, etc.
- Components with full MCDA methods
- Visualization components
- Reporting/comparison components

What is nice about diviz (so far)?

- Access to **multiple methods**
- **Interface and logic is the same**, although methods may differ a lot
- **Construction** of MCDA workflows (=methods) from elementary components

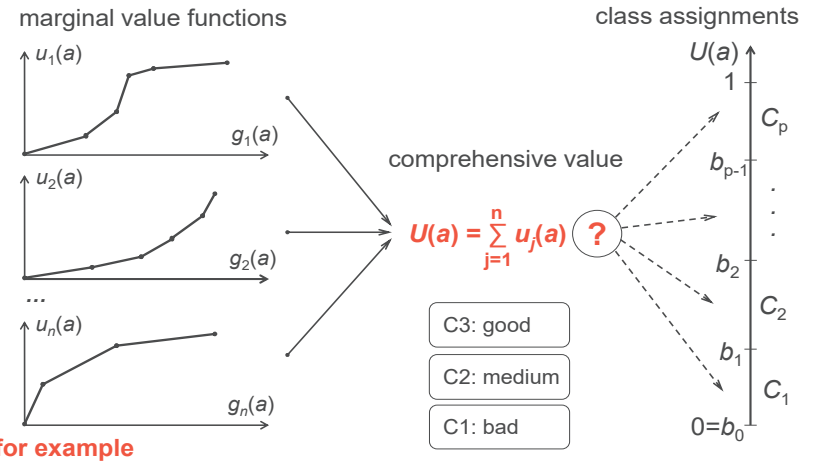
- **Comparing logic and outcomes of different approaches**
  - compare rankings obtained with different methods by **visual means**
    or with **Kendall's coefficient**
    how many pairwise comparisons agreed/not?
    1 – full agreement, -1 – disagreement
- **Easy to prepare input and share output**
  - workflow: import / export options

1. a04	1. a12
2. a12	2. a04
3. a02	3. a02

diviz demo (3)

TIME FOR DEMO
COMPARING RESULTS
OF DIFFERENT METHODS

Threshold-based value driven sorting



for example

- The lower and upper threshold for class *medium* (C2) are 0.4 and 0.7.
- If $U(a10) = 0.5$, $(0.4 \leq 0.5 < 0.7)$, it would be assigned to class medium.

Basic Robust Ordinal Regression for sorting

preference information

assignment examples
- *a12* should be assigned to class *good*: *a12* → C3
- *a04* should not be assigned to class *bad*: *a04* → [C2, C3]

preference model

set of all value functions and class thresholds compatible with DM's preference

exploitation with linear programming

results

assignments
- necessary assignment confirmed by all compatible models
- possible assignment confirmed by at least one compatible model

Different types of input preference information

assignment examples

- *a12* should be assigned to class *good: a12* → C3
- *a04* should not be assigned to class *bad: a04* → [C2, C3]

assignment-based pairwise comparisons

- *a03* is better than *a05* by at least one *class*
- The class difference between *a07* and *a01* is at most one
- *a11* and *a12* should be assigned to the same *class*

desired class cardinalities

- At most 5 cars can be assigned to class *good*
- At least 40% of cars should be assigned to class *bad*
- The number of cars assigned to class *medium* should be between 3 and 7

Different types of output sorting results

assignment examples
assignment-based pair-wise comparisons
desired class cardinalities

preference information

set of all value functions and class thresholds compatible with DM's preference

exploitation with linear programming

recommendation

assignments **assignment-based preference relations** **class cardinalities**

variety of results

necessary = for all, possible = for at least one
extreme = the most and the least advantageous

for example:

- necessary **assignment-based preference relation**:
  a05 is necessarily **assigned to a class at least as good** as *a06*
- **extreme class cardinalities**:
  the **minimal/maximal number** of cars assigned to class medium is 5

Preference information reflected in results

a12 should be assigned to class *good: a12* → C3 — **INPUT**

a12 is necessarily assigned to class *good: a12* $\rightarrow^{N}$ C3 — **OUTPUT**

a11 and *a12* should be assigned to the same *class* — **INPUT**

a11 is necessarily assigned to a class at least as good as *a12*
a12 is necessarily assigned to a class at least as good as *a11* — **OUTPUT**

The number of cars assigned to medium should be between 3 and 7 — **INPUT**

The minimal number of cars assigned to class medium will be ≥ 3
The maximal number of cars assigned to class medium will be ≤ 7 — **OUTPUT**

Results motivate enrichment of preference information

Possible assignment too wide — **OUTPUT**

Add more precise assignment example — **INPUT**

Alternatives incomparable in terms of the necessary assignment-based relation

Add additional assignment-based pairwise comparison — **INPUT**

Class cardinalities very imprecise — **OUTPUT**

Add more precise requirements on desired class cardinalities — **INPUT**

EVOLUTION OF RESULTS WITH GROWTH OF PREFERENCE INFORMATION

- Possible assignments become more precise
- Necessary assignment-based preference relation is enriched
- Extreme class cardinalities get closer to each other

diviz demo (4)

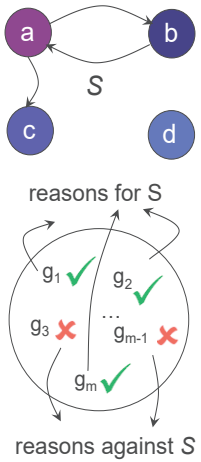
TIME FOR DEMO

ROR-UTADIS

Outranking preference model

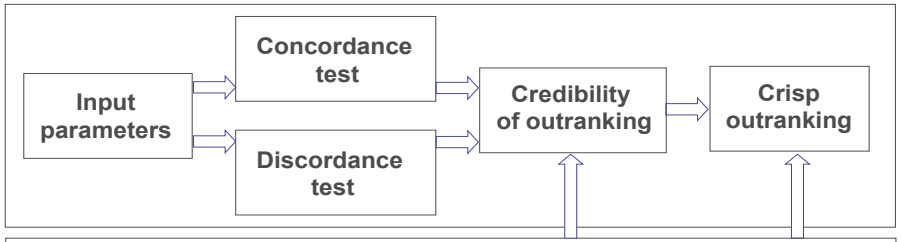
- **a outranks b** (a*S*b) if the arguments of a decision maker in favour of the statement "**a is at least as good as b**" are strong enough and there arguments oppposite to this statements are weak

- These arguments are based on:
  - The **evaluations** of a and b on the various criteria
  - **Information on the preference** of the decision maker: criteria weight (w_j), indifference (q_j), preference (p_j), pre-veto (discordance) (pv_j) and veto (v_j) thresholds for each criterion, and cutting level (λ)
- **Remark**: if no argument can be found neither in favour of aSb nor in favour of bSa → **incomparability**



ELECTRE methods: step by step

Construction of an outranking relation



Exploitation of outranking relation in a way specific for ranking, choice or sorting

CHOICE
- ELECTRE I
- ELECTRE Iv
- ELECTRE Is
- …

RANKING
- ELECTRE II
- ELECTRE III
- ELECTRE IV
- …

SORTING
- ELECTRE TRI-B
- ELECTRE TRI-C
- ELECTRE TRI-rC
- MR-SORT
- THESEUS

Partial concordance indices

Compute partial concordance index for each pair of alternatives

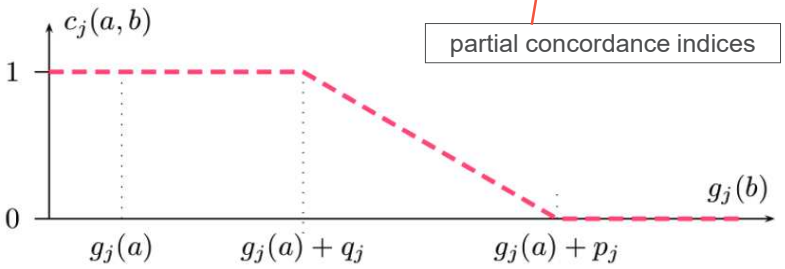
indifference threshold on critetion g_1

for example, if $g_1(a04) - g_1(a05) \geq -q_1$ ⇒ $c_1(a04,a05) = 1.0$

for example, if $g_2(a05) - g_2(a04) \leq -p_2$ ⇒ $c_2(a04,a05) = 0.0$

preference threshold on critetion g_2

partial concordance indices



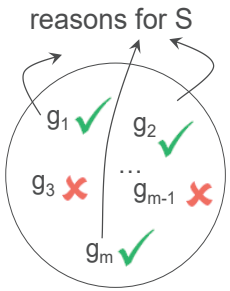
Comprehensive concordance index

Compute comprehensive concordance index for each pair of objects:

the contribution of all criteria to the proposition aSb

$$C(a,b) = \sum_j w_j \cdot c_j(a,b) = w_1 \cdot c_1(a,b) + w_2 \cdot c_2(a,b) + \ldots + w_n \cdot c_n(a,b)$$

weight associated with criterion g_j
$\sum_{j=1\ldots n} w_j = 1$

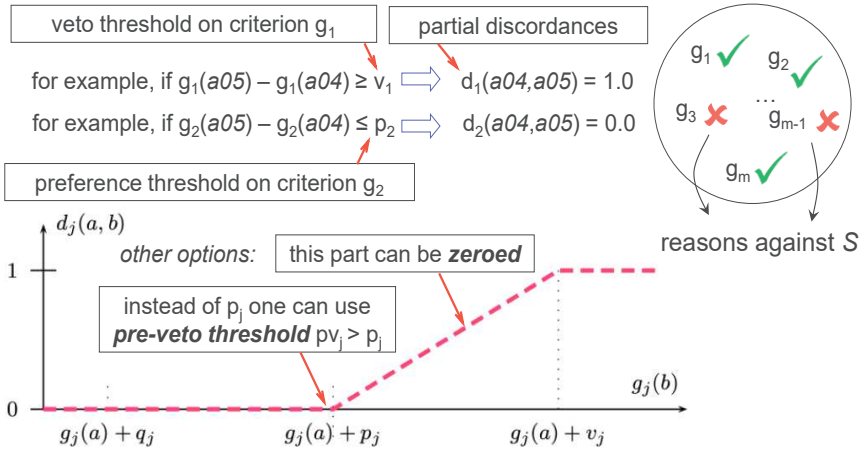


More adavanced options account for:

- *Interactions* between criteria
  (mutual strengthening, mutual weakening, antagonistic effect)
- *Reinforced preference* effect (very strong reasons for *S*)

Partial discordance indices

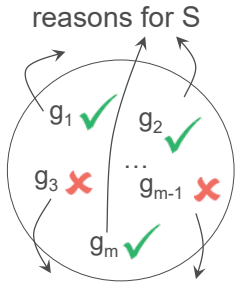
- **Compute partial discordance**: measures the degree to which a criterion is discordant (i.e., express opposition) with the proposition a*S*b



Credbility = valued outranking relation

The (valued) outranking relation can be defined by a **credibility index** σ(a,b):

- if no criterion is discordant: σ(a,b) = C(a,b)
- if at least one criterion is discordant: σ(a,b) < C(a,b)
- if $d_j(a,b)=1$ for at least one criterion: σ(a,b) = 0



other options (no denominator):

Formulation:

$$\sigma(a,b) = C(a,b) \prod_{j \in F} \frac{1-d_j(a,b)}{1-C(a,b)}$$

where $F = \{j : d_j(a,b) > C(a,b)\}$

instead of all sufficiently great arguments against outranking, account only for **the greatest (max) one** (not product, but max)

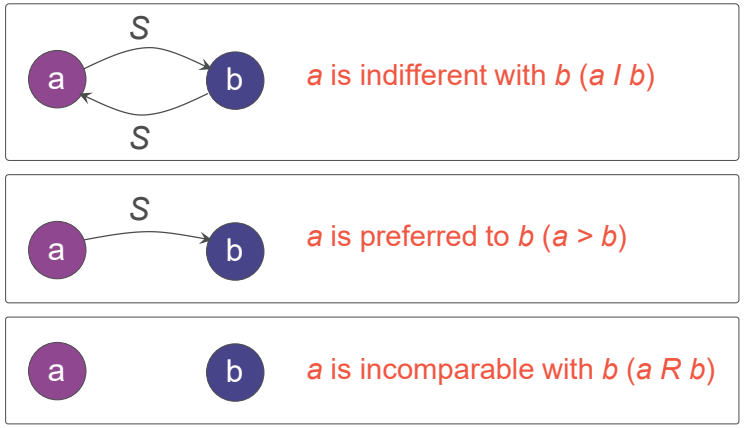
instead of all **sufficiently great** arguments against outranking, include **all** arguments againts (no j∈F)

also computable without weights as in ELECTRE IV

Crisp outranking relation

Comparison of a credibility index with **cutting level λ** (is it high enough?)

$$\sigma(a,b) \geq \lambda \Rightarrow aSb$$



Net Flow Score procedure

NFS(a) = strength(a) – weakness(a)

- exploitation of a **valued** outranking relation

$$NFS^{\sigma}(a) = \sum_{b \in A} [\sigma(a,b) - \sigma(b,a)]$$

quality | strength | weakness

σ(a,b)	a	b	c
a	-	0.8	0.4
b	0.6	-	0.7
c	0.3	0.4	-

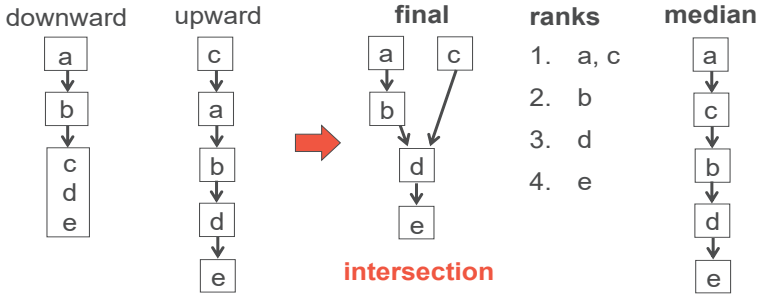
$NFS^{Sval}(a)$ = 1.2 - 0.9 = 0.3

$NFS^{Sval}(b)$ = 1.3 - 1.2 = 0.1

$NFS^{Sval}(c)$ = 0.7 - 1.1 = -0.4

ELECTRE III - distillation procedures

- distillation procedure exploiting a valued outranking relation
- **downward** pre-order (constructed top-down)
  - identify alternatives A_1 with the greatest quality
  - put A_1 at the top, and continue with A/A_1, etc.
- **upward** pre-order (constructed bottom-up)
  - identify alternatives A_1 with the least quality
  - put A_1 at the bottom, and continue with A/A_1, etc.

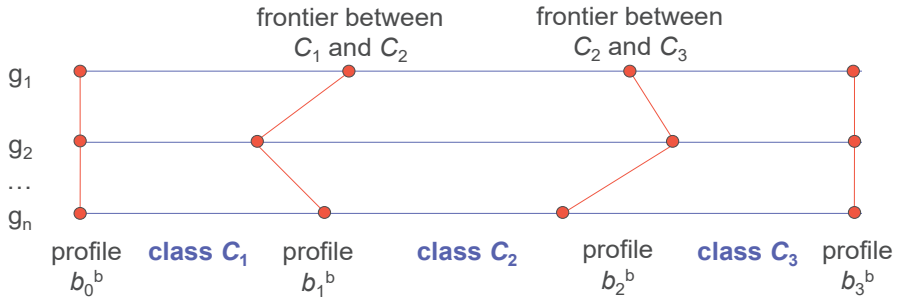


diviz demo (5)

TIME FOR DEMO

RANKING WITH ELECTRE

ELECTRE TRI-B

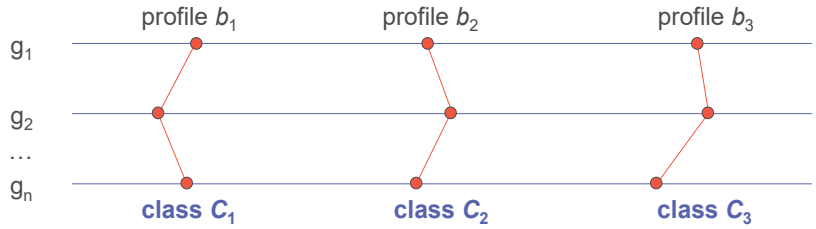


- **boundary profiles** used for modeling the frontiers between classes
- two **disjoint assigment rules** for assignment of alternative *a*

pessimistic rule
- start from the best profile
- find the first profile b_h^b : $a\ S\ b_h^b$
- select C_{h+1}

optimistic rule
- start from the worst profile
- find the first profile b_h^b : $b_h^b > a$
- select C_h

ELECTRE TRI-rC



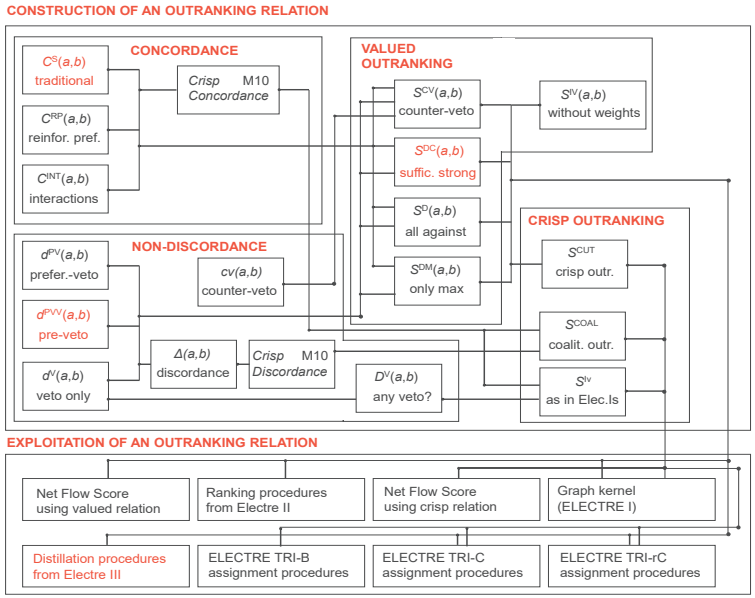
- **characteristic profiles** formed from the **class representative criteria values**
- two **conjoint assigment rules** for assignment of alternative *a* indicating:
- **the worst class of *a***
  - start from the second best profile
  - find the first profile b_h:
    $a > b_h$ and $\sigma(a,b_{h+1}) > \sigma(b_h,a)$
  - select C_{h+1}
- **the best class of *a***
  - start from the second worst profile
  - find the first profile b_h:
    $b_h > a$ and $\sigma(b_{h-1},a) > \sigma(a,b_h)$
  - select C_{h-1}
- indications of these two rules combined into a recommended class interval

diviz demo (6)

TIME FOR DEMO

SORTING WITH ELECTRE

Construct your own ELECTRE



diviz in 2018

What *diviz* is?

- tool for MCDA component workflow
- simple data visualization tool
- platform independent and open source

UTA, UTASTAR, **ACUTA**, UTAMP, Robust Ordinal Regression (UTA-GMS), RUTA, Extreme Ranking Analysis, SMAA-2, Stochastic Ordinal Regression

ROR-UTADIS (including UTADIS-GMS)

"**Construct your own Electre**"

"**Construct your own Promethee**"

over 1000 variants of Electre and Promethee

Aggregation operators: **weighted sum**, OWA, Choquet integral, etc.

Rubis, MR-Sort, clustering

Visualisation, descriptive stats, reports, **comparison methods**, and many many more ☺

Data Envelopment Analysis: CCR and value-based model

robust and stochastic analysis

Summary (1)

Make MCDA software publicly available

- "I like the procedure described in this paper, where can I test it?"
- Both the traditional methods and brand new ones

Decompose the MCDA methods into elementary components

- Give the possibility to create workflow of such components
- MCDA methods, algorithmic components and data visualization modules are available as web services
- Components can interoperate via the XMCDA standard

Have you ever wished what would happen if…?
How do the results of one methods differ from these of another one?

Expect more from us….
…on both visual and methodological sites

Summary (2)

How you can help the project?

- Join the Decision Deck Consortium
- Test the software & send us your opinion
- Let us know what you need

Important websites

- http://www.decision-deck.org
- http://www.diviz.org, @divizMCDA, +diviz - all information on *diviz*
- getting help: http://www.diviz.org/contact
  S. Bigaret, P. Meyer, M. Kadziński, V. Mousseau, M. Pirlot, …

R. Bisdorff, L. Dias, P. Meyer, V. Mousseau, M. Pirlot, *Evaluation and Decision Models with Multiple Criteria*, Springer, 2015

Hands on exercises

- **Construction** of some of the previously presented "method" in diviz
- Help Thierry to **choose the car** which is "best" for him
- 2 roles in each group:
  - The **analyst** constructs the MCDA algorithmic workflows
  - The **decision maker** (Thierry) is questioned by the analyst on his/her preferences

Practical work (see detailed instructions)

- **Multi-Attribute Value Theory (steps V)**
- UTA (steps U)
- **Electre III (steps E)**
- **Promethee (steps P)**
- Comparing results (steps C)