Supporting collaborative multi-criteria evaluation: the VIP Analysis plug-in for Decision Deck

João N. Clímaco, João A. Costa, Luis C. Dias, Paulo Melo

INESC Coimbra, Portugal

Fac. Economia, Univ. Coimbra, Portugal

GDN 2008 Group Decision and Negotiation Meeting
Coimbra, Portugal, June 17-20, 2008

Plan of the presentation

- **Introduction to VIP Analysis**
  - Methodology
  - Original software implementation
- **The VIP Analysis plug-in for Decision Deck 1.1**
  - Brief Introduction
  - Short demonstration
- **Vip Analysis (D2 vs Delphi version)**

Introduction to VIP Analysis

Introduction to VIP Analysis 1:

What is VIP Analysis?

- VIP Analysis is an existing multicriteria DSS for choice problems
- based on additive utility (or value) functions

$$V(a_x) = k_1.v_1(a_x) + k_2.v_2(a_x) + \ldots + k_n.v_n(a_x)$$

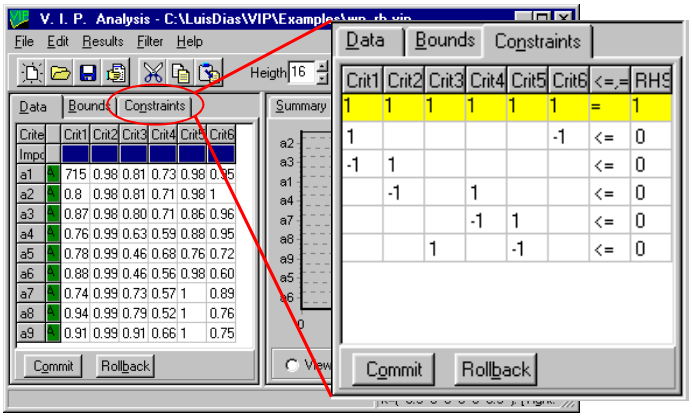
- which accepts imprecise information on the scaling coefficients associated with the criteria, e.g., a ranking of these coefficients

$$k_1 \geq k_2 \geq \ldots \geq k_n$$

- or any other linear constraints, including holistic comparisons of alternatives, trade-off bounds, etc. to define a domain T of acceptable parameter values.
- Rather than selecting the centroid $(k_1,\ldots,k_n)^*$ of T...
- it computes the range of results compatible with that information, to look for conclusions that hold for every $(k_1,\ldots,k_n)^* \in T$ ("robust conclusions").

Introduction to VIP Analysis 2:

VIP Analysis Inputs

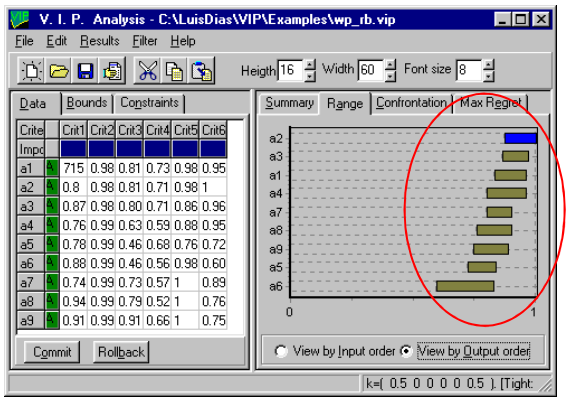


Introduction to VIP Analysis 3:

VIP Analysis Outputs

Minimum/maximum value for each alternative.

min $\{V(a_x) : (k_1,\ldots,k_n)\in T\}$, *max* $\{V(a_x) : (k_1,\ldots,k_n)\in T\}$.



Introduction to VIP Analysis 4:

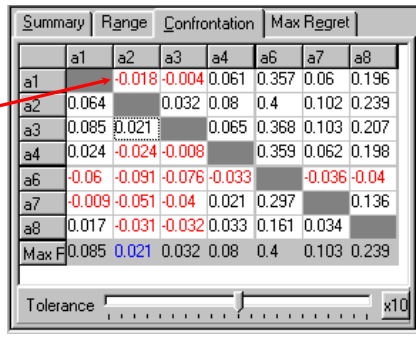
VIP Analysis Outputs

Maximum advantage of a_x over a_y

$M_{xy} = max\ \{V(a_x) - V(a_y): (k_1,\ldots,k_n) \in T\}$.

→ *Relative dominance*
(e.g., a_1 dominated by a_2)

→ $max\ \{M_{xy} : y \neq x\} \leq 0 \Leftrightarrow$
$\Leftrightarrow$ a_y is "optimal"



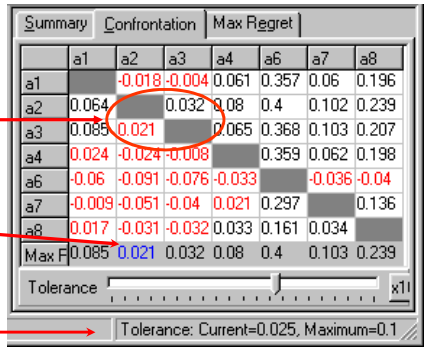
Introduction to VIP Analysis 5:

VIP Analysis Outputs

Tolerances w.r.t. maximum advantage of a_x over a_y.

→ Δ_ε : Quasi-dominance
(e.g., $a_2\ \Delta_{0.025}\ a_3$)

→ $max\ \{M_{xy} : y \neq x\} \leq \varepsilon \Leftrightarrow$
$\Leftrightarrow$ a_y is "quasi-optimal"



ε

Introduction to VIP Analysis 6:

VIP Analysis use

- Robust conclusions:
  - allow DMs to eliminate (quasi-) dominated alternatives,
  - and perhaps select a (quasi-) optimal one.
- Information on variability of results:
  - to know which alternatives are more affected by imprecision,
  - to guide the elicitation of further information,
  - to suggest further constraints on T.
- Therefore, allows DMs to learn about the model and the problem,
- and to postpone elicitation questions they find difficult

Introduction to VIP Analysis 7:

VIP Analysis Software

- Programmed in Borland Delphi in the late 1990s for Windows
- Distributed as a freeware (without sources) to those who request it
- requested by 150+ users from several countries, besides Portugal: Argentina, Australia, Austria, Bangladesh, Belgium, Brazil, Canada, Chile, China & Taiwan, Colombia, Costa Rica, Cuba, Czech Rep., Ecuador, Egypt, Finland, France, Greece, Hungary, Indonesia, Iran, Italia, Japan, Lithuania, Malaysia, Maroc, The Netherlands, New Zealand, Poland, Rumania, Russia, Slovenia, Spain, Switzerland, UK, USA, Venezuela and Vietnam.
- References:

Dias, L. C. and J. N. Clímaco, Additive Aggregation with Variable Interdependent Parameters: the VIP Analysis Software, *Journal of the Operational Research Society* 51 (9), 1070-1082, 2000.

Extension: Dias, L.C., J.N. Clímaco, Dealing with imprecise information in group multicriteria decisions: A methodology and a GDSS architecture, *European Journal of Operational Research* 160 (2), 291-307, 2005.

http://www4.fe.uc.pt/lmcdias/english/vipa.htm

The VIP Analysis plug-in for Decision Deck 1.1

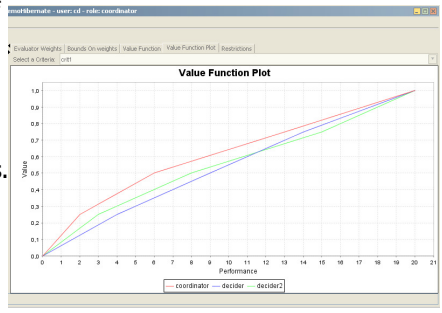
(evolution of v1.0 to Hibernate)

Brief Introduction 1:

User roles: Coordinator, Evaluator, DM

- The Coordinator role encompasses the roles of Administrator, Alternatives designer, and Criteria designer.
- As a Coordinator, the user can create a new problem, setting the problem's name, the names of the alternatives, and the names of the criteria, besides defining who are the other users and their roles.
- A Coordinator <u>can</u> define the shape of value functions or allow the DMs to define their individual value functions.
- A Coordinator <u>can</u> define weights for the evaluators or allow the DMs to define their individual evaluator weights.

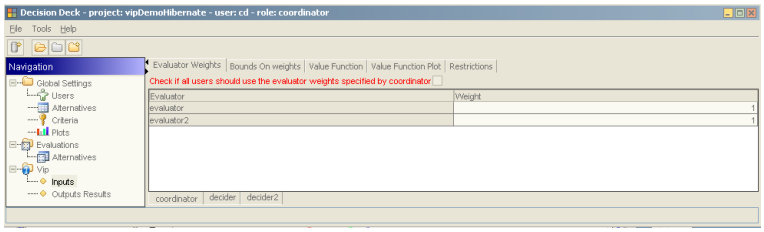
⇒ the Coordinator can ensure that all DMs work with the same value scales and performances.



Brief Introduction 2:

User roles: Coordinator, Evaluator, DM

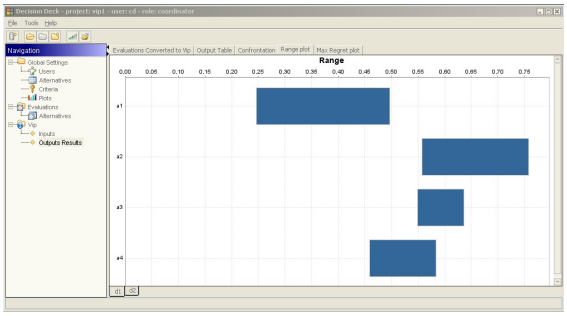
- as an Evaluator, the user can only indicate the performance of the each alternative on different criteria, according to his or her expertise.
- Since not all evaluators are supposed to be experts for all the evaluation criteria, an evaluator can leave the evaluations of the alternatives void on some criteria.
- A virtual evaluator is defined by a weighted average, using weights (set by the Coordinator or a DM) reflecting different degrees of confidence in the evaluations.



Brief Introduction 3:

User roles: Coordinator, Evaluator, DM

- as a DM, the user can specify value functions (if the Coordinator allows)
- and can also specify weights for the evaluators (again, if the Coordinator allows).
- The DM is the only role that can specify the method's preference-related parameters, i.e. the scaling weights $(w_1,\ldots,w_n)$ and/or constraints.
- Can compute outputs.
- Can submit outputs.



Demonstration

VIP Analysis

(D2 Version Vs Delphi Version)

Software Comparison:

Gains and losses

+ Distributed and concurrent access over a network
+ Different user roles
+ Specification of value functions by coordinator or DMs
+ Open source philosophy
+ Integration with other methods in D^2 within a coherent interface
- Installation requirements and procedure
- Some of the original functionalities were abandoned (but of little use)
- Saving, exporting, or importing data (but could be programmed)

Software Comparison:

Future

- D2 makes sense in multi-actor settings
- Currently difficult to use as a GDSS: lack of communication and **awareness** mechanisms (Who is logged in? What are they doing? Why? What happened since I last logged out?, ...)
  - Web services for coordination
    - bonus: browser-based interfacing
  - Traceability could also play an importat role