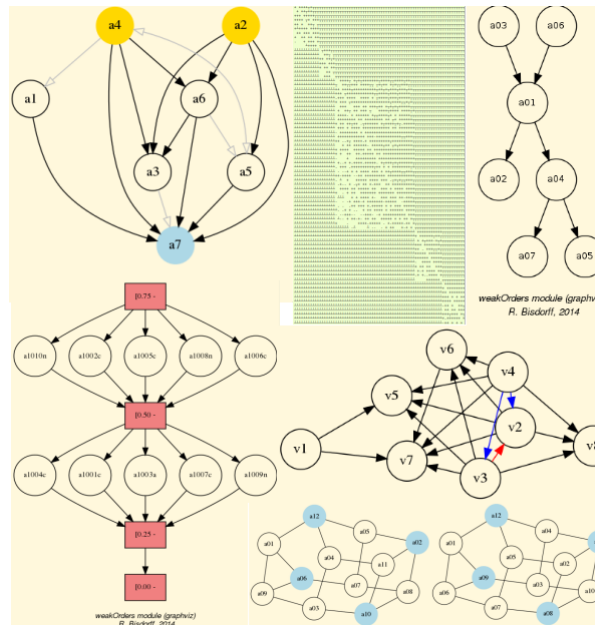


Documentation of the DIGRAPH3 software collection



Tutorials and Advanced Topics

Raymond BISOORFF

Luxembourg, 2020

Last updated : October 4, 2021

*This documentation is dedicated to our
late colleague and dear friend
Prof. Marc ROUBENS.*

More documents are freely available [here](#)

Contents

1	Coping with missing data and indeterminateness	2
1.1	A motivating data set	3
1.2	Modelling pairwise bipolar-valued rating opinions	6
2	Ordinal correlation equals bipolar-valued relational equivalence	8
2.1	Kendall's <i>tau</i> index	9
2.2	Bipolar-valued relational equivalence	10
2.3	Fitness of ranking heuristics	12
2.4	Illustrating preference divergences	14
2.5	Exploring the <i>better rated</i> and the <i>as well as rated</i> opinions	16
3	Bipolar-valued kernel membership characteristic vectors	18
3.1	Kernel equation systems	19
3.2	Solving bipolar-valued kernel equation systems	20
4	On confident outrankings with uncertain criteria significances	27
4.1	Modelling uncertain criteria significances	27
4.2	Bipolar-valued likelihood of 'at least as good as' situations	28
4.3	Confidence level of outranking situations	31
5	On stable outrankings with ordinal criteria significances	36
5.1	Cardinal or ordinal criteria significances	37
5.2	Qualifying the stability of outranking situations	38
5.3	Computing the stability denotation of outranking situations	43
5.4	Robust bipolar-valued outranking digraphs	44
6	On unopposed outrankings with multiple decision objectives	48
6.1	Characterising unopposed multiobjective outranking situations	48
6.2	Computing <i>Pareto</i> efficient multiobjective choices	51
7	Two-stage elections with multipartisan primary selection	53
7.1	Converting voting profiles into performance tableaux	53
7.2	Multipartisan primary selection of eligible candidates	55
7.3	Secondary election winner determination	57
7.4	Multipartisan preferences in divisive politics	58
8	Tempering plurality tyranny effects with bipolar approval voting	62
8.1	Bipolar approval voting systems	63
8.2	Pairwise comparison of bipolar approval votes	66
8.3	Three-valued evaluative voting system	68
8.4	Favouring multipartisan candidates	71
9	Bibliography	76
	References	76

Digraph3 Advanced Topics

HTML Version

“The rule for the combination of independent concurrent arguments takes a very simple form when expressed in terms of the intensity of belief ... It is this: Take the sum of all the feelings of belief which would be produced separately by all the arguments pro, subtract from that the similar sum for arguments con, and the remainder is the feeling of belief which ought to have the whole. This is a proceeding which men often resort to, under the name of balancing reasons.”

—C.S. Peirce, The probability of induction (1878)

In this part of the **Digraph3** *documentation*, we provide an insight in computational enhancements one may get when working in a *bipolar-valued logical framework*, like easily coping with *missing data* and uncertain criterion *significance weights*, computing valued *ordinal correlations* between bipolar-valued outranking digraphs, solving bipolar-valued kernel equation systems, and testing for stability and confidence of outranking statements when facing uncertain performance criteria significances or decision objectives’ importance weights.

1 Coping with missing data and indeterminateness

- *A motivating data set* (page 3)
- *Modelling pairwise bipolar-valued rating opinions* (page 6)

In a stubborn keeping with a two-valued logic, where every argument can only be true or false, there is no place for efficiently taking into account missing data or logical indeterminateness. These cases are seen as problematic and, at best are simply ignored. Worst, in modern data science, missing data get often replaced with *fictive* values, potentially falsifying hence all subsequent computations.

In social choice problems like elections, *abstentions* are, however, frequently observed and represent a social expression that may be significant for revealing non represented social preferences.

In marketing studies, interviewees will not always respond to all the submitted questions. Again, such abstentions do sometimes contain nevertheless valid information concerning consumer preferences.

1.1 A motivating data set

Let us consider such a performance tableau in file `graffiti07.py` gathering a *Movie Magazine* 's rating of some movies that could actually be seen in town¹ (see Fig. 1.1).

```
1 >>> from outrankingDigraphs import *
2 >>> t = PerformanceTableau('graffiti07')
3 >>> t.showHTMLPerformanceTableau(title='Graffiti Star wars',\
4 ...                               ndigits=0)
```

Graffiti Star wars

criteria	AP	AS	CF	CS	DR	FG	GS	JH	JPT	MR	RR	SF	SJ	TD	VT
weight	1.00	1.00	1.00	1.00	1.00	1.00	1.00	3.00	1.00	1.00	1.00	1.00	1.00	1.00	3.00
mv_AL	3	-1	2	-1	NA	NA	3	NA	2	NA	NA	NA	2	NA	2
mv_BI	1	-1	1	1	-1	NA	NA	NA	2	NA	2	NA	NA	NA	NA
mv_CM	NA	3	3	2	NA	NA	3	2	3	2	1	2	2	NA	2
mv_DF	NA	4	3	1	2	NA	1	3	2	2	2	3	-1	NA	1
mv_DG	3	2	2	3	3	1	4	3	3	3	NA	NA	-1	NA	3
mv_DI	1	2	NA	1	2	2	NA	1	2	2	NA	1	1	NA	NA
mv_DJ	3	1	3	NA	NA	NA	3	NA	2	2	NA	2	4	3	-1
mv_FC	3	2	2	1	3	NA	1	3	3	3	2	NA	2	1	3
mv_FF	2	3	2	1	2	2	NA	1	2	2	1	2	1	3	1
mv_GG	2	3	3	1	NA	NA	1	-1	NA	-1	2	NA	-1	NA	1
mv_GH	1	NA	1	-1	2	2	1	2	1	3	4	NA	NA	NA	NA
mv_HN	3	1	3	1	3	NA	4	3	2	NA	1	NA	3	3	3
mv_HP	1	NA	3	1	NA	NA	NA	1	1	2	3	NA	1	2	NA
mv_HS	2	4	2	3	2	2	2	3	4	2	3	NA	1	3	NA
mv_JB	3	4	3	NA	3	2	3	2	3	2	NA	2	2	NA	3
mv_MB	NA	2	NA	NA	1	NA	1	2	2	1	2	NA	2	2	NA
mv_NP	NA	1	3	1	NA	3	2	3	3	3	2	3	NA	NA	3
mv_PE	3	4	2	NA	3	1	NA	2	4	2	NA	3	3	NA	3
mv_QS	NA	3	2	NA	4	3	NA	4	4	3	3	4	NA	4	4
mv_RG	2	2	2	2	NA	NA	3	1	2	2	1	NA	NA	3	NA
mv_RR	3	2	NA	1	4	NA	4	3	3	4	3	3	NA	4	2
mv_SM	3	3	2	2	2	2	NA	3	2	2	3	NA	2	2	3
mv_TF	-1	NA	1	1	1	NA	NA	1	2	NA	-1	NA	-1	1	NA
mv_TM	2	1	2	2	NA	NA	2	2	2	3	NA	2	NA	4	NA
mv_TP	2	3	3	1	2	NA	NA	3	2	NA	2	NA	1	NA	2

Fig. 1.1: *Graffiti* magazine's movie ratings from September 2007

15 journalists and movie critics provide here their rating of 25 movies: 5 stars (*master-piece*), 4 stars (*must be seen*), 3 stars (*excellent*), 2 stars (*good*), 1 star (*could be seen*), -1 star (*I do not like*), -2 (*I hate*), NA (*not seen*).

To aggregate all the critics' rating opinions, the *Graffiti* magazine provides for each movie a global score computed as an *average grade*, just ignoring the *not seen* data. These

¹ *Graffiti*, Edition Revue Luxembourg, September 2007, p. 30. You may find the data file *graffiti07.py* (perfTabs.PerformanceTableau Format) in the *examples* directory of the Digraph3 ressources.

averages are thus not computed on comparable denominators; some critics do indeed use a more or less extended range of grades. The movies not seen by critic *SJ*, for instance, are favored, as this critic is more severe than others in her grading. Dropping the movies that were not seen by all the critics is here not possible either, as no one of the 25 movies was actually seen by all the critics. Providing any value for the missing data will as well always somehow falsify any global value scoring. What to do ?

A better approach is to rank the movies on the basis of pairwise bipolar-valued *at least as well rated as* opinions. Under this epistemic argumentation approach, missing data are naturally treated as opinion abstentions and hence do not falsify the logical computations. Such a ranking (see the tutorial on Ranking with incommensurable performance criteria) of the 25 movies is provided, for instance, by the **heat map** view shown in [Fig. 1.2](#).

```
>>> t.showHTMLPerformanceHeatmap(Correlations=True,\
...                               rankingRule='NetFlows',
...                               ndigits=0)
```

Ranking the movies

criteria	JH	JPT	AP	DR	MR	VT	GS	CS	SJ	RR	TD	CF	SF	AS	FG
weights	+3.00	+1.00	+1.00	+1.00	+1.00	+3.00	+1.00	+1.00	+1.00	+1.00	+1.00	+1.00	+1.00	+1.00	+1.00
tau(*)	+0.50	+0.43	+0.32	+0.26	+0.25	+0.23	+0.16	+0.14	+0.14	+0.13	+0.11	+0.11	+0.10	+0.08	+0.03
mv_QS	4	4	NA	4	3	4	NA	NA	NA	3	4	2	4	3	3
mv_RR	3	3	3	4	4	2	4	1	NA	3	4	NA	3	2	NA
mv_DG	3	3	3	3	3	3	4	3	-1	NA	NA	2	NA	2	1
mv_NP	3	3	NA	NA	3	3	2	1	NA	2	NA	3	3	1	3
mv_HN	3	2	3	3	NA	3	4	1	3	1	3	3	NA	1	NA
mv_HS	3	4	2	2	2	NA	2	3	1	3	3	2	NA	4	2
mv_SM	3	2	3	2	2	3	NA	2	2	3	2	2	NA	3	2
mv_JB	2	3	3	3	2	3	3	NA	2	NA	NA	3	2	4	2
mv_PE	2	4	3	3	2	3	NA	NA	3	NA	NA	2	3	4	1
mv_FC	3	3	3	3	3	3	1	1	2	2	1	2	NA	2	NA
mv_TP	3	2	2	2	NA	2	NA	1	1	2	NA	3	NA	3	NA
mv_CM	2	3	NA	NA	2	2	3	2	2	1	NA	3	2	3	NA
mv_DF	3	2	NA	2	2	1	1	1	-1	2	NA	3	3	4	NA
mv_TM	2	2	2	NA	3	NA	2	2	NA	NA	4	2	2	1	NA
mv_DJ	NA	2	3	NA	2	-1	3	NA	4	NA	3	3	2	1	NA
mv_AL	NA	2	3	NA	NA	2	3	-1	2	NA	NA	2	NA	-1	NA
mv_RG	1	2	2	NA	2	NA	3	2	NA	1	3	2	NA	2	NA
mv_MB	2	2	NA	1	1	NA	1	NA	2	2	2	NA	NA	2	NA
mv_GH	2	1	1	2	3	NA	1	-1	NA	4	NA	1	NA	NA	2
mv_HP	1	1	1	NA	2	NA	NA	1	1	3	2	3	NA	NA	NA
mv_BI	NA	2	1	-1	NA	NA	NA	1	NA	2	NA	1	NA	-1	NA
mv_DI	1	2	1	2	2	NA	NA	1	1	NA	NA	NA	1	2	2
mv_FF	1	2	2	2	2	1	NA	1	1	1	3	2	2	3	2
mv_GG	-1	NA	2	NA	-1	1	1	1	-1	2	NA	3	NA	3	NA
mv_TF	1	2	-1	1	NA	NA	NA	1	-1	-1	1	1	NA	NA	NA

Color legend:

quantile	20.00%	40.00%	60.00%	80.00%	100.00%
----------	--------	--------	--------	--------	---------

(*) tau: Ordinal (Kendall) correlation between marginal criterion and global ranking relation

Outranking model: **standard**, Ranking rule: **NetFlows**

Ordinal (Kendall) correlation between global ranking and global outranking relation: **+0.780**

Mean marginal correlation (a) : **+0.234**

Standard marginal correlation deviation (b) : **+0.147**

Ranking fairness (a) - (b) : **+0.087**

Fig. 1.2: *Graffiti* magazine's ordered movie ratings from September 2007

There is no doubt that movie *mv_QS*, with 6 'must be seen' marks, is correctly best-ranked and the movie *mv_TV* is worst-ranked with five 'don't like' marks.

1.2 Modelling pairwise bipolar-valued rating opinions

Let us explicitly construct the underlying bipolar-valued outranking digraph and consult in Fig. 1.3 the pairwise characteristic values we observe between the two best-ranked movies, namely *mv_QS* and *mv_RR*.

```
1 >>> g = BipolarOutrankingDigraph(t)
2 >>> g.recodeValuation(-19,19) # integer characteristic values
3 >>> g.showHTMLPairwiseOutrankings('mv_QS','mv_RR')
```

Pairwise Comparison											
Comparing actions : (mv_QS,mv_RR)											
crit.	wght.	g(x)	g(y)	diff	ind	wp	p	concord	wv	v	polarisation
AP	1.00	NA	3.00					0.00			
AS	1.00	3.00	2.00	+1.00	None	None	None	+1.00			
CF	1.00	2.00	NA					0.00			
CS	1.00	NA	1.00					0.00			
DR	1.00	4.00	4.00	+0.00	None	None	None	+1.00			
FG	1.00	3.00	NA					0.00			
GS	1.00	NA	4.00					0.00			
JH	3.00	4.00	3.00	+1.00	None	None	None	+3.00			
JPT	1.00	4.00	3.00	+1.00	None	None	None	+1.00			
MR	1.00	3.00	4.00	-1.00	None	None	None	-1.00			
RR	1.00	3.00	3.00	+0.00	None	None	None	+1.00			
SF	1.00	4.00	3.00	+1.00	None	None	None	+1.00			
SJ	1.00	NA	NA					0.00			
TD	1.00	4.00	4.00	+0.00	None	None	None	+1.00			
VT	3.00	4.00	2.00	+2.00	None	None	None	+3.00			
Valuation in range: -19.00 to +19.00; global concordance: +11.00											

Pairwise Comparison											
Comparing actions : (mv_RR,mv_QS)											
crit.	wght.	g(x)	g(y)	diff	ind	wp	p	concord	wv	v	polarisation
AP	1.00	3.00	NA					0.00			
AS	1.00	2.00	3.00	-1.00	None	None	None	-1.00			
CF	1.00	NA	2.00					0.00			
CS	1.00	1.00	NA					0.00			
DR	1.00	4.00	4.00	+0.00	None	None	None	+1.00			
FG	1.00	NA	3.00					0.00			
GS	1.00	4.00	NA					0.00			
JH	3.00	3.00	4.00	-1.00	None	None	None	-3.00			
JPT	1.00	3.00	4.00	-1.00	None	None	None	-1.00			
MR	1.00	4.00	3.00	+1.00	None	None	None	+1.00			
RR	1.00	3.00	3.00	+0.00	None	None	None	+1.00			
SF	1.00	3.00	4.00	-1.00	None	None	None	-1.00			
SJ	1.00	NA	NA					0.00			
TD	1.00	4.00	4.00	+0.00	None	None	None	+1.00			
VT	3.00	2.00	4.00	-2.00	None	None	None	-3.00			
Valuation in range: -19.00 to +19.00; global concordance: -5.00											

Fig. 1.3: Pairwise comparison of the two best-ranked movies

Six out of the fifteen critics have not seen one or the other of these two movies. Notice the higher significance (3) that is granted to two locally renowned movie critics, namely *JH* and *VT*. Their opinion counts for three times the opinion of the other critics. All nine critics that have seen both movies, except critic *MR*, state that *mv_QS* is rated at least as well as *mv_RR* and the balance of positive against negative opinions amounts to +11, a characteristic value which positively validates the outranking situation with a majority of $(11/19 + 1.0) / 2.0 = 79\%$.

The complete table of pairwise majority margins of global ‘at least as well rated as’ opinions, ranked by the same rule as shown in the heat map above (see Fig. 1.2), may be shown in Fig. 1.4.

```
1 >>> ranking = g.computeNetFlowsRanking()
2 >>> g.showHTMLRelationTable(actionsList=ranking, ndigits=0,\
3 ...     tableTitle='Bipolar characteristic values of\
4 ...     "rated at least as good as" situations')
```

Bipolar characteristic values of "rated at least as good as" situations

r(x S y)	mv_QS	mv_RR	mv_DG	mv_NP	mv_HN	mv_HS	mv_SM	mv_JB	mv_PE	mv_FC	mv_TP	mv_CM	mv_DF	mv_TM	mv_DJ	mv_AL	mv_RG	mv_MB	mv_GH	mv_HP	mv_BI	mv_DI	mv_FF	mv_GG	mv_FF
mv_QS	-	11	12	11	10	9	14	9	11	13	9	10	9	9	7	6	9	9	7	6	5	9	15	8	8
mv_RR	-5	-	5	7	8	6	4	5	2	9	10	9	12	9	10	8	9	10	8	9	6	10	13	10	9
mv_DG	-8	9	-	9	10	5	9	7	8	13	7	7	10	10	6	8	10	7	9	5	6	7	9	9	9
mv_NP	-7	5	7	-	10	3	7	9	8	11	9	8	12	8	7	6	4	6	8	6	5	7	12	10	7
mv_HN	-10	4	8	8	-	5	9	9	8	10	10	7	10	6	8	10	7	6	8	8	5	7	13	9	11
mv_HS	-3	2	5	5	3	-	10	2	5	6	9	5	10	7	0	1	10	9	8	9	7	11	14	9	11
mv_SM	-8	6	7	5	7	6	-	6	6	10	12	9	10	6	4	9	9	10	7	9	7	11	15	11	11
mv_JB	-9	-1	5	1	3	4	8	-	9	6	6	13	6	8	9	9	9	9	8	8	5	11	15	12	8
mv_PE	-7	2	6	0	4	3	6	11	-	5	4	10	5	7	6	8	8	8	5	6	5	9	13	9	8
mv_FC	-9	5	11	9	8	2	8	8	7	-	10	6	11	5	3	8	6	9	9	5	7	10	12	10	11
mv_TP	-7	4	-1	3	2	3	0	-4	-4	0	-	6	11	6	4	5	7	6	7	7	7	9	14	12	10
mv_CM	-8	-1	-5	-4	-3	-3	-1	5	-2	-2	4	-	3	8	8	9	10	7	5	7	3	9	14	11	8
mv_DF	-9	-2	-2	2	2	2	2	0	1	1	5	1	-	4	6	-1	6	8	6	5	6	8	13	13	9
mv_TM	-3	-7	-6	-2	-2	-1	0	0	1	-1	-2	2	-2	-	2	2	7	6	9	7	5	7	9	5	8
mv_DJ	-7	4	-4	-3	2	4	2	1	-2	-1	0	0	0	4	-	3	5	4	3	6	4	4	3	1	5
mv_AL	-4	0	-4	-6	-6	1	-1	-3	-4	-2	3	1	3	2	3	-	2	2	5	1	3	1	5	3	3
mv_RG	-7	-7	-6	-4	-1	-2	-1	-5	-4	-2	-3	-2	-2	1	3	4	-	1	0	6	4	8	9	4	9
mv_MB	-9	-8	-5	-4	-4	-9	-4	-1	-2	-1	-2	1	-2	2	-2	2	3	-	2	4	4	4	2	6	8
mv_GH	-5	-8	-7	-6	-8	-4	-3	0	1	-5	-5	1	-2	-1	-3	-3	0	6	-	5	2	5	3	3	5
mv_HP	-4	-5	-5	-2	-4	-3	-3	-4	-4	-3	-1	-3	1	-7	-2	-1	2	-2	-1	-	3	6	5	7	8
mv_BI	-5	-4	-6	-1	-1	-7	-5	-5	-3	-1	-3	0	-3	-2	1	-2	0	2	1	-	1	-1	-1	4	4
mv_DI	-9	-6	-5	-5	-3	-3	-3	-7	-7	-6	-1	-7	0	-5	0	1	4	0	1	8	5	-	6	4	8
mv_FF	-11	-11	-7	-10	-5	0	-3	-9	-9	-8	-2	-6	1	-1	5	-1	9	0	1	7	5	12	-	9	11
mv_GG	-6	-8	-7	-4	-5	-7	-9	-10	-9	-4	-2	-7	3	-3	1	-3	-2	-2	-1	-3	5	-2	3	-	2
mv_FF	-8	-7	-7	-5	-7	-11	-9	-8	-8	-7	-6	-8	-3	-6	-3	-1	-1	-4	-3	0	2	2	-1	2	-

Valuation domain: [-19.00; +19.00]

Fig. 1.4: Pairwise majority margins of ‘at least as well rated as’ rating opinions

Positive characteristic values, validating a global ‘at least as well rated as’ opinion are marked in light green (see Fig. 1.4). Whereas negative characteristic values, invalidating such a global opinion, are marked in light red. We may by the way notice that the best-ranked movie *mv_QS* is indeed a *Condorcet* winner, i.e. *better rated than all the other movies* by a 65% majority of critics. This majority may be assessed from the average determinateness of the given bipolar-valued outranking digraph *g*.

```
>>> print( '%.0f%%' % g.computeDeterminateness(InPercents=True) )
65%
```

Notice also the *indeterminate* situation we observe, for instance, when comparing movie *mv_PE* with movie *mv_NP*.

```
>>> g.showHTMLPairwiseComparison('mv_PE', 'mv_NP')
```


Pairwise Comparison

Comparing actions : (mv_PE,mv_NP)

crit.	wght.	g(x)	g(y)	diff	ind	wp	p	concord	wv	v	polarisation
AP	1.00	3.00	NA					0.00			
AS	1.00	4.00	1.00	+3.00	None	None	None	+1.00			
CF	1.00	2.00	3.00	-1.00	None	None	None	-1.00			
CS	1.00	NA	1.00					0.00			
DR	1.00	3.00	NA					0.00			
FG	1.00	1.00	3.00	-2.00	None	None	None	-1.00			
GS	1.00	NA	2.00					0.00			
JH	3.00	2.00	3.00	-1.00	None	None	None	-3.00			
JPT	1.00	4.00	3.00	+1.00	None	None	None	+1.00			
MR	1.00	2.00	3.00	-1.00	None	None	None	-1.00			
RR	1.00	NA	2.00					0.00			
SF	1.00	3.00	3.00	+0.00	None	None	None	+1.00			
SJ	1.00	3.00	NA					0.00			
TD	1.00	NA	NA					0.00			
VT	3.00	3.00	3.00	+0.00	None	None	None	+3.00			

Valuation in range: -19.00 to +19.00; global concordance: +0.00

Fig. 1.5: Indeterminate pairwise comparison example

Only eight, out of the fifteen critics, have seen both movies and the positive opinions do neatly balance the negative ones. A global statement that *mv_PE* is ‘at least as well rated as’ *mv_NP* may in this case hence **neither be validated, nor invalidated**; a preferential situation that cannot be modelled with any scoring approach.

It is fair, however, to eventually mention here that the *Graffiti* magazine’s average scoring method is actually showing a very similar ranking. Indeed, average scores usually confirm well all evident pairwise comparisons, yet *enforce* comparability for all less evident ones.

Notice finally the ordinal correlation *tau* values in Fig. 1.2 3rd row. How may we compute these ordinal correlation indexes ?

Back to *Content Table* (page 2)

2 Ordinal correlation equals bipolar-valued relational equivalence

- *Kendall’s tau index* (page 9)
- *Bipolar-valued relational equivalence* (page 10)
- *Fitness of ranking heuristics* (page 12)
- *Illustrating preference divergences* (page 14)
- *Exploring the better rated and the as well as rated opinions* (page 16)

2.1 Kendall's *tau* index

M. G. Kendall ([KEN-1938p]) defined his *ordinal correlation* τ (**tau**) *index* for linear orders of dimension n as a *balancing* of the number $\#Co$ of correctly oriented pairs against the number $\#In$ of incorrectly oriented pairs. The total number of irreflexive pairs being $n(n-1)$, in the case of linear orders, $\#Co + \#In = n(n-1)$. Hence $\tau = \left(\frac{\#Co}{n(n-1)}\right) - \left(\frac{\#In}{n(n-1)}\right)$. In case $\#In$ is zero, $\tau = +1$ (all pairs are *equivalently oriented*); inversely, in case $\#Co$ is zero, $\tau = -1$ (all pairs are *differently oriented*).

Noticing that $\frac{\#Co}{n(n-1)} = 1 - \frac{\#In}{n(n-1)}$, and recalling that the bipolar-valued negation is operated by changing the sign of the characteristic value, *Kendall's* original *tau* definition implemented in fact the bipolar-valued **negation** of the **non equivalence** of two linear orders:

$$\tau = 1 - 2\frac{\#In}{n(n-1)} = -\left(2\frac{\#In}{n(n-1)} - 1\right) = 2\frac{\#Co}{n(n-1)} - 1,$$

i.e. the **normalized majority margin** of *equivalently oriented* irreflexive pairs.

Let $R1$ and $R2$ be two random crisp relations defined on a same set of 5 alternatives. We may compute Kendall's *tau* index as follows.

Listing 2.1: Crisp Relational Equivalence Digraph

```

1  >>> from digraphs import *
2  >>> R1 = RandomDigraph(order=5,Bipolar=True)
3  >>> R2 = RandomDigraph(order=5,Bipolar=True)
4  >>> E = EquivalenceDigraph(R1,R2)
5  >>> E.showRelationTable(ReflexiveTerms=False)
6  * ---- Relation Table ----
7  r(<=>)|  'a1'      'a2'      'a3'      'a4'      'a5'
8  -----|-----
9  'a1' |   -        -1.00    1.00   -1.00    1.00
10 'a2' | -1.00         -    -1.00    1.00   -1.00
11 'a3' | -1.00      -1.00     -    1.00    1.00
12 'a4' | -1.00        1.00   -1.00     -    1.00
13 'a5' | -1.00        1.00   -1.00    1.00     -
14 Valuation domain: [-1.00;1.00]
15 >>> E.correlation
16 {'correlation': -0.1, 'determination': 1.0}

```

In the table of the equivalence relation ($R_1 \Leftrightarrow R_2$) above (see Listing 2.1 Lines 9-13), we observe that the normalized majority margin of equivalent versus non equivalent irreflexive pairs amounts to $(9 - 11)/20 = -0.1$, i.e. the value of Kendall's *tau* index in this plainly determined crisp case (see Listing 2.1 Line 16).

What happens now with more or less determined and even partially indeterminate relations ? May we proceed in a similar way ?

2.2 Bipolar-valued relational equivalence

Let us now consider two randomly bipolar-valued digraphs $R1$ and $R2$ of order five.

Listing 2.2: Two Random Bipolar-valued Digraphs

```

1 >>> R1 = RandomValuationDigraph(order=5,seed=1)
2 >>> R1.showRelationTable(ReflexiveTerms=False)
3 * ---- Relation Table ----
4 r(R1)| 'a1'      'a2'      'a3'      'a4'      'a5'
5 -----|-----
6 'a1' | -         -0.66     0.44     0.94     -0.84
7 'a2' | -0.36      -         -0.70     0.26     0.94
8 'a3' | 0.14       0.20      -         0.66     -0.04
9 'a4' | -0.48      -0.76     0.24      -         -0.94
10 'a5' | -0.02      0.10      0.54     0.94      -
11 Valuation domain: [-1.00;1.00]
12 >>> R2 = RandomValuationDigraph(order=5,seed=2)
13 >>> R2.showRelationTable(ReflexiveTerms=False)
14 * ---- Relation Table ----
15 r(R2)| 'a1'      'a2'      'a3'      'a4'      'a5'
16 -----|-----
17 'a1' | -         -0.86     -0.78     -0.80     -0.08
18 'a2' | -0.58      -         0.88      0.70     -0.22
19 'a3' | -0.36      0.54      -         -0.46     0.54
20 'a4' | -0.92      0.48      0.74      -         -0.60
21 'a5' | 0.10       0.62      0.00      0.84      -
22 Valuation domain: [-1.00;1.00]

```

We may notice in the relation tables shown above that 9 pairs, like $(a1,a2)$ or $(a3,a2)$ for instance, appear equivalently oriented (see Listing 2.2 Lines 6,17 or 8,19). The `EquivalenceDigraph` class implements this *relational equivalence* relation between digraphs $R1$ and $R2$ (see Listing 2.3).

Listing 2.3: Bipolar-valued Equivalence Digraph

```

1 >>> eq = EquivalenceDigraph(R1,R2)
2 >>> eq.showRelationTable(ReflexiveTerms=False)
3 * ---- Relation Table ----
4 r(<=>)| 'a1'      'a2'      'a3'      'a4'      'a5'
5 -----|-----
6 'a1' | -         0.66     -0.44     -0.80     0.08
7 'a2' | 0.36      -         -0.70     0.26     -0.22
8 'a3' | -0.14      0.20      -         -0.46     -0.04
9 'a4' | 0.48      -0.48     0.24      -         0.60
10 'a5' | -0.02      0.10      0.00      0.84      -
11 Valuation domain: [-1.00;1.00]

```

In our bipolar-valued epistemic logic, logical disjunctions and conjunctions are implemented as *max*, respectively *min* operators. Notice also that the logical equivalence

$(R_1 \Leftrightarrow R_2)$ corresponds to a double implication $(R_1 \Rightarrow R_2) \wedge (R_2 \Rightarrow R_1)$ and that the implication $(R_1 \Rightarrow R_2)$ is logically equivalent to the disjunction $(\neg R_1 \vee R_2)$.

When $r(x R_1 y)$ and $r(x R_2 y)$ denote the bipolar-valued characteristic values of relation R_1 , resp. R_2 , we may hence compute as follows a majority margin $M(R_1 \Leftrightarrow R_2)$ between equivalently and not equivalently oriented irreflexive pairs (x, y) .

$$M(R_1 \Leftrightarrow R_2) = \sum_{(x \neq y)} \left[\min \left(\max(-r(x R_1 y), r(x R_2 y)), \max(-r(x R_2 y), r(x R_1 y)) \right) \right].$$

$M(R_1 \Leftrightarrow R_2)$ is thus given by the sum of the non reflexive terms of the relation table of *eq*, the relation equivalence digraph computed above (see Listing 2.3).

In the crisp case, $M(R_1 \Leftrightarrow R_2)$ is now normalized with the maximum number of possible irreflexive pairs, namely $n(n-1)$. In a generalized r -valued case, the maximal possible equivalence majority margin M corresponds to the sum D of the **conjoint determinations** of $(x R_1 y)$ and $(x R_2 y)$ (see [BIS-2012p]).

$$D = \sum_{x \neq y} \min \left[\text{abs}(r(x R_1 y)), \text{abs}(r(x R_2 y)) \right].$$

Thus, we obtain in the general r -valued case:

$$\tau(R_1, R_2) = \frac{M(R_1 \Leftrightarrow R_2)}{D}.$$

$\tau(R_1, R_2)$ corresponds thus to a classical ordinal correlation index, but restricted to the **conjointly determined parts** of the given relations R_1 and R_2 . In the limit case of two crisp linear orders, D equals $n(n-1)$, i.e. the number of irreflexive pairs, and we recover hence *Kendall* 's original *tau* index definition.

It is worthwhile noticing that the ordinal correlation index $\tau(R_1, R_2)$ we obtain above corresponds to the ratio of

$$\begin{aligned} r(R_1 \Leftrightarrow R_2) &= \frac{M(R_1 \Leftrightarrow R_2)}{n(n-1)} : \text{the normalized majority margin of the pairwise} \\ &\text{relational equivalence statements, also called } \textit{valued ordinal correlation}, \text{ and} \\ d &= \frac{D}{n(n-1)} : \text{the normalized determination of the corresponding pairwise} \\ &\text{relational equivalence statements, in fact the } \textit{determinateness} \text{ of the relational} \\ &\text{equivalence digraph.} \end{aligned}$$

We have thus successfully **out-factored** the *determination* effect from the *correlation* effect. With completely determined relations, $\tau(R_1, R_2) = r(R_1 \Leftrightarrow R_2)$. By convention, we set the ordinal correlation with a *completely indeterminate* relation, i.e. when $D = 0$, to the *indeterminate* correlation value 0.0. With *uniformly* chosen random r -valued relations, the **expected** *tau* index is **0.0**, denoting in fact an **indeterminate** correlation. The corresponding expected normalized determination d is about 0.333 (see [BIS-2012p]).

We may verify these relations with help of the corresponding equivalence digraph *eq* (see Listing 2.4).

Listing 2.4: Computing the Ordinal Correlation Index
from the Equivalence Digraph

```
1 >>> eq = EquivalenceDigraph(R1,R2)
2 >>> M = Decimal('0'); D = Decimal('0')
```

(continues on next page)

(continued from previous page)

```
3 >>> n2 = eq.order*(eq.order - 1)
4 >>> for x in eq.actions:
5     ...     for y in eq.actions:
6     ...         M += eq.relation[x][y]
7     ...         D += abs(eq.relation[x][y])

1 >>> print('r(R1<=>R2) = %+.3f, d = %.3f, tau = %+.3f' % (M/n2,D/n2,M/D))
2 r(R1<=>R2) = +0.026, d = 0.356, tau = +0.073
```

In general we simply use the `computeOrdinalCorrelation()` method which renders a dictionary with a ‘*correlation*’ (*tau*) and a ‘*determination*’ (*d*) attribute. We may recover $r(<=>)$ by multiplying *tau* with *d* (see Listing 2.5 Line 4).

Listing 2.5: Directly Computing the Ordinal Correlation Index

```
1 >>> corrR1R2 = R1.computeOrdinalCorrelation(R2)
2 >>> tau = corrR1R2['correlation']
3 >>> d = corrR1R2['determination']
4 >>> r = tau * d
5 >>> print('tau(R1,R2) = %+.3f, d = %.3f,\
6 ...     r(R1<=>R2) = %+.3f' % (tau, d, r))
7
8 tau(R1,R2) = +0.073, d = 0.356, r(R1<=>R2) = +0.026
```

We provide for convenience a direct `showCorrelation()` method:

```
1 >>> corrR1R2 = R1.computeOrdinalCorrelation(R2)
2 >>> R1.showCorrelation(corrR1R2)
3 Correlation indexes:
4 Extended Kendall tau      : +0.073
5 Epistemic determination  : 0.356
6 Bipolar-valued equivalence : +0.026
```

We may now illustrate the quality of the global ranking of the movies shown with the heat map in Fig. 1.2.

2.3 Fitness of ranking heuristics

We reconsider the bipolar-valued outranking digraph g modelling the pairwise global ‘*at least as well rated as*’ relation among the 25 movies seen above (see Listing 2.6).

Listing 2.6: Global Movies Outranking Digraph

```
1 >>> g = BipolarOutrankingDigraph(t,Normalized=True)
2 *----- Object instance description -----*
3 Instance class      : BipolarOutrankingDigraph
```

(continues on next page)

(continued from previous page)

```
4 Instance name      : rel_grafittiPerfTab.xml
5 # Actions          : 25
6 # Criteria         : 15
7 Size               : 390
8 Determinateness    : 65%
9 Valuation domain  : {'min': Decimal('-1.0'),
10                      'med': Decimal('0.0'),
11                      'max': Decimal('1.0'),}
12 >>> g.computeCoSize()
13 188
```

Out of the $25 \times 24 = 600$ irreflexive movie pairs, digraph g contains 390 positively validated, 188 positively invalidated, and 22 *indeterminate* outranking situations (see the zero-valued cells in Fig. 1.4).

Let us now compute the normalized majority margin $r(<=>)$ of the equivalence between the marginal critic's pairwise ratings and the global *Net-Flows* ranking shown in the ordered heat map (see Fig. 1.2).

Listing 2.7: Marginal Criterion Correlations with global *NetFlows* Ranking

```
1 >>> from linearOrders import NetFlowsOrder
2 >>> nf = NetFlowsOrder(g)
3 >>> nf.netFlowsRanking
4 ['mv_QS', 'mv_RR', 'mv_DG', 'mv_NP', 'mv_HN', 'mv_HS', 'mv_SM',
5  'mv_JB', 'mv_PE', 'mv_FC', 'mv_TP', 'mv_CM', 'mv_DF', 'mv_TM',
6  'mv_DJ', 'mv_AL', 'mv_RG', 'mv_MB', 'mv_GH', 'mv_HP', 'mv_BI',
7  'mv_DI', 'mv_FF', 'mv_GG', 'mv_TF']
8 >>> for i,item in enumerate(\
9     g.computeMarginalVersusGlobalRankingCorrelations(\
10     nf.netFlowsRanking,ValuedCorrelation=True) ):\
11     print('r(%s<=>nf) = %.3f' % (item[1],item[0]) )
12
13 r(JH<=>nf) = +0.500
14 r(JPT<=>nf) = +0.430
15 r(AP<=>nf) = +0.323
16 r(DR<=>nf) = +0.263
17 r(MR<=>nf) = +0.247
18 r(VT<=>nf) = +0.227
19 r(GS<=>nf) = +0.160
20 r(CS<=>nf) = +0.140
21 r(SJ<=>nf) = +0.137
22 r(RR<=>nf) = +0.133
23 r(TD<=>nf) = +0.110
24 r(CF<=>nf) = +0.110
25 r(SF<=>nf) = +0.103
```

(continues on next page)

(continued from previous page)

```
26 r(AS<=>nf) = +0.080
27 r(FG<=>nf) = +0.027
```

In Listing 2.7 (see Lines 13-27), we recover above the relational equivalence characteristic values shown in the third row of the table in Fig. 1.2. The global *Net-Flows* ranking represents obviously a rather balanced compromise with respect to all movie critics' opinions as there appears no valued negative correlation with anyone of them. The *Net-Flows* ranking apparently takes also correctly in account that the journalist *JH*, a locally renowned movie critic, shows a higher significance weight (see Line 13).

The ordinal correlation between the global *Net-Flows* ranking and the digraph *g* may be furthermore computed as follows:

Listing 2.8: Correlation between outrankings global *Net-Flows* Ranking

```
1 >>> corrgnf = g.computeOrdinalCorrelation(nf)
2 >>> g.showCorrelation(corrgnf)
3 Correlation indexes:
4 Extended Kendall tau      : +0.780
5 Epistemic determination   :  0.300
6 Bipolar-valued equivalence : +0.234
```

We notice in Listing 2.8 Line 4 that the ordinal correlation $\tau(g, nf)$ index between the *Net-Flows* ranking *nf* and the determined part of the outranking digraph *g* is quite high (+0.78). Due to the rather high number of missing data, the *r*-valued relational equivalence between the *nf* and the *g* digraph, with a characteristics value of *only* +0.234, may be misleading. Yet, +0.234 still corresponds to an epistemic majority support of nearly 62% of the movie critics' rating opinions.

It would be interesting to compare similarly the correlations one may obtain with other global ranking heuristics, like the *Copeland* or the *Kohler* ranking rule.

2.4 Illustrating preference divergences

The valued relational equivalence index gives us a further measure for studying how **divergent** appear the rating opinions expressed by the movie critics.

```

1 >>> g = BipolarOutrankingDigraph(t,Normalized=True)
2 >>> g.showCriteriaCorrelationTable(ValuedCorrelation=True)
3 Criteria valued ordinal correlation index
4
5 -----
6 AP | +0.63 +0.04 +0.19 +0.09 +0.22 -0.01 +0.11 +0.23 +0.25 +0.08 +0.02 +0.04 +0.19 +0.04 +0.12
7 AS |      +0.77 +0.12 +0.12 +0.04 -0.02 -0.06 +0.02 +0.24 -0.08 +0.07 +0.04 -0.07 -0.01 +0.02
8 CF |          +0.77 +0.07 +0.11 +0.03 +0.05 +0.07 +0.10 -0.03 +0.01 +0.00 +0.06 +0.03 -0.04
9 CS |              +0.63 +0.04 -0.02 +0.07 +0.13 +0.25 +0.01 +0.03 +0.00 +0.02 +0.03 +0.07
10 DR |                  +0.45 +0.03 +0.07 +0.17 +0.23 +0.16 +0.06 +0.03 +0.10 +0.07 +0.10
11 FG |                      +0.15 -0.01 +0.04 +0.01 +0.06 -0.00 +0.02 +0.01 +0.01 +0.02
12 GS |                          +0.40 +0.07 +0.07 +0.09 -0.02 +0.00 +0.06 +0.04 +0.04
13 JH |                              +0.77 +0.28 +0.26 +0.15 +0.12 +0.10 +0.05 +0.14
14 JPT |                                  +0.92 +0.15 +0.06 +0.09 +0.08 +0.08 +0.17
15 MR |                                      +0.63 +0.10 +0.08 +0.03 +0.09 +0.10
16 RR |                                          +0.51 +0.04 +0.01 +0.05 +0.05
17 SF |                                              +0.18 +0.01 +0.02 +0.05
18 SJ |                                                  +0.51 +0.03 +0.07
19 TD |                                                      +0.26 +0.00
20 VT |                                                          +0.40

```

Fig. 2.1: Pairwise valued correlation of movie critics

It is remarkable to notice in the criteria correlation matrix (see Fig. 2.1) that, due to the quite numerous missing data, all pairwise valued ordinal correlation indexes $r(x \leq y)$ appear to be of low value, except the *diagonal* ones. These reflexive indexes $r(x \leq x)$ would trivially all amount to +1.0 in a plainly determined case. Here they indicate a reflexive normalized determination score d , i.e. the *proportion* of pairs of movies each critic did evaluate. Critic *JPT* (the editor of the Graffiti magazine), for instance, evaluated all but one ($d = 24 \cdot 23 / 600 = 0.92$), whereas critic *FG* evaluated only 10 movies among the 25 in discussion ($d = 10 \cdot 9 / 600 = 0.15$).

To get a picture of the actual *divergence of rating opinions* concerning **jointly seen** pairs of movies, we may develop a *Principal Component Analysis* ⁽²⁾ of the corresponding *tau* correlation matrix. The 3D plot of the first 3 principal axes is shown in Fig. 2.2.

```
>>> g.export3DplotOfCriteriaCorrelation(ValuedCorrelation=False)
```

² The 3D PCA plot method requires a running *R statistics software* (<https://www.r-project.org/>) installation and the Calmat matrix calculator (see the calmat directory in the Digraph3 ressources)

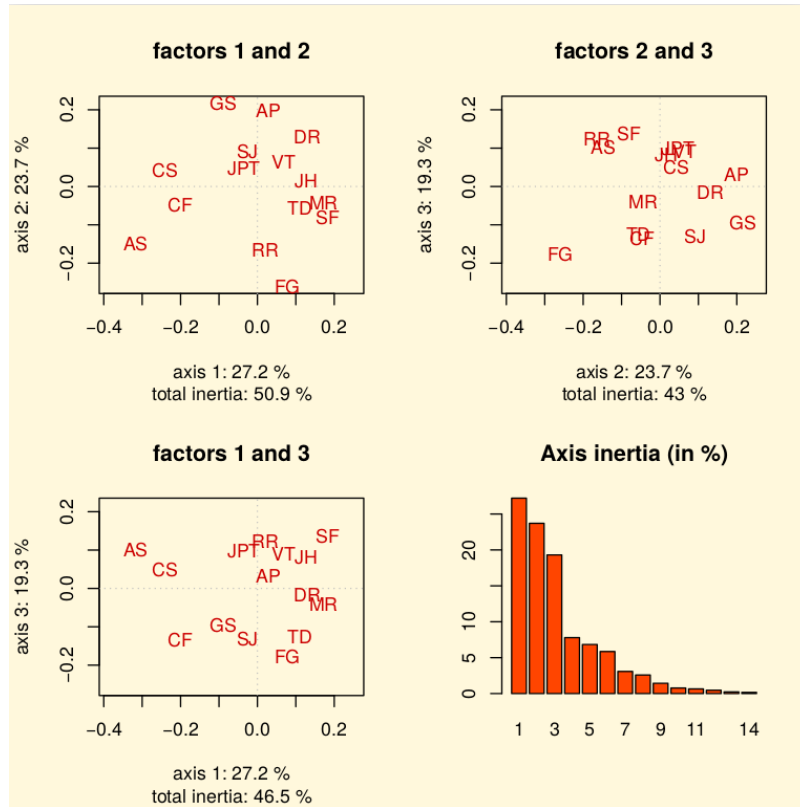


Fig. 2.2: 3D PCA plot of the criteria ordinal correlation matrix

The first 3 principal axes support together about 70% of the total inertia. Most *eccentric* and *opposed* in their respective rating opinions appear, on the first principal axis with 27.2% inertia, the conservative daily press against labour and public press. On the second principal axis with 23.7.7% inertia, it is the people press versus the cultural critical press. And, on the third axis with still 19.3% inertia, the written media appear most opposed to the radio media.

2.5 Exploring the *better rated* and the *as well as rated* opinions

In order to furthermore study the quality of a ranking result, it may be interesting to have a separate view on the asymmetric and symmetric parts of the ‘*at least as well rated as*’ opinions (see the tutorial on Manipulating Digraph objects).

Let us first have a look at the pairwise asymmetric part, namely the ‘*better rated than*’ and ‘*less well rated than*’ opinions of the movie critics.

```
>>> ag = AsymmetricPartialDigraph(g)
>>> ag.showHTMLRelationTable(actionsList=g.computeNetFlowsRanking(),
  ↪ndigits=0)
```

Valued Adjacency Matrix

r(x S y)	mv_QS	mv_RR	mv_DG	mv_NP	mv_HN	mv_HS	mv_SM	mv_JB	mv_PE	mv_FC	mv_TP	mv_CM	mv_DF	mv_TM	mv_DJ	mv_AL	mv_RG	mv_MB	mv_GH	mv_HP	mv_BI	mv_DI	mv_FF	mv_GG	mv_TF
mv_QS	-	11	12	11	10	9	14	9	11	13	9	10	9	9	7	6	9	9	7	6	5	9	15	8	8
mv_RR	-5	-	0	0	0	0	0	5	0	0	0	9	12	9	10	8	9	10	8	9	6	10	13	10	9
mv_DG	-8	0	-	0	0	0	0	0	0	0	7	7	10	10	6	8	10	7	9	5	6	7	9	9	9
mv_NP	-7	0	0	-	0	0	0	0	8	0	0	8	0	8	7	6	4	6	8	6	5	7	12	10	7
mv_HN	-10	0	0	0	-	0	0	0	0	0	0	7	0	6	0	10	7	6	8	8	5	7	13	9	11
mv_HS	-3	0	0	0	0	-	0	0	0	0	0	5	0	7	0	0	10	9	8	9	7	11	14	9	11
mv_SM	-8	0	0	0	0	0	-	0	0	0	12	9	0	6	0	9	9	10	7	9	7	11	15	11	11
mv_JB	-9	-1	0	0	0	0	0	-	0	0	6	0	6	8	0	9	9	9	8	8	5	11	15	12	8
mv_PE	-7	0	0	0	0	0	0	0	-	0	4	10	0	0	6	8	8	8	0	6	5	9	13	9	8
mv_FC	-9	0	0	0	0	0	0	0	0	-	10	6	0	5	3	8	6	9	9	5	7	10	12	10	11
mv_TP	-7	0	-1	0	0	0	0	-4	-4	0	-	0	0	6	4	0	7	6	7	7	7	9	14	12	10
mv_CM	-8	-1	-5	-4	-3	-3	-1	0	-2	-2	0	-	0	0	8	0	10	0	0	7	3	9	14	11	8
mv_DF	-9	-2	-2	0	0	0	0	0	0	0	0	0	-	4	6	-1	6	8	6	0	6	8	0	0	9
mv_TM	-3	-7	-6	-2	-2	-1	0	0	0	-1	-2	0	-2	-	0	0	0	0	9	7	5	7	9	5	8
mv_DJ	-7	-8	-4	-3	0	4	0	0	-2	-1	0	0	0	-	0	0	4	3	6	4	4	0	0	5	3
mv_AL	-4	0	-4	-6	-6	0	-1	-3	-4	-2	0	0	3	0	0	-	0	5	1	0	0	5	3	0	3
mv_RG	-7	-7	-6	-4	-1	-2	-1	-5	-4	-2	-3	-2	-2	0	0	0	-	0	0	0	4	0	0	4	9
mv_MB	-9	-8	-5	-4	-4	-9	-4	-1	-2	-1	-2	0	-2	0	0	-	0	0	4	4	4	2	6	8	8
mv_GH	-5	-8	-7	-6	-8	-4	-3	0	0	-5	-5	0	-2	-1	-3	-3	0	0	-	5	0	0	0	3	5
mv_HP	-4	-5	-5	-2	-4	-3	-3	-4	-4	-3	-1	-3	0	-7	-2	-1	0	-2	-1	-	0	0	0	7	8
mv_BI	-5	-4	-6	-1	-1	-7	-5	-5	-5	-3	-1	-3	0	-2	0	-2	0	0	0	0	-	0	-1	-1	0
mv_DI	-9	-6	-5	-5	-3	-3	-3	-7	-7	-6	-1	-7	0	-5	0	0	0	0	0	0	0	-	0	4	0
mv_FF	-11	-11	-7	-10	-5	0	-3	-9	-9	-8	-2	-6	0	-1	0	-1	0	0	0	5	0	-	0	11	0
mv_GG	-6	-8	-7	-4	-5	-7	-9	-10	-9	-4	-2	-7	0	-3	0	-3	-2	-2	-1	-3	5	-2	0	-	0
mv_TF	-8	-7	-7	-5	-7	-11	-9	-8	-8	-7	-6	-8	-3	-6	-3	-1	-1	-4	-3	0	0	0	-1	0	-

Valuation domain: [-19.00; +19.00]

Fig. 2.3: Asymmetric part of graffiti07 digraph

We notice here that the *Net-Flows* ranking rule inverts in fact just three ‘*less well ranked than*’ opinions and four ‘*better ranked than*’ ones. A similar look at the symmetric part, the pairwise ‘*as well rated as*’ opinions, suggests a preordered preference structure in several *equivalently rated* classes.

```
>>> sg = SymmetricPartialDigraph(g)
>>> sg.showHTMLRelationTable(actionsList=g.computeNetFlowsRanking(),
    ↪ndigits=0)
```

Valued Adjacency Matrix

r(x S y)	mv_QS	mv_RR	mv_DG	mv_NP	mv_HN	mv_HS	mv_SM	mv_JB	mv_PE	mv_FC	mv_TP	mv_CM	mv_DF	mv_TM	mv_DJ	mv_AL	mv_RG	mv_MB	mv_GH	mv_HP	mv_BI	mv_DI	mv_FF	mv_GG	mv_TF
mv_QS	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mv_RR	0	-	5	7	8	6	4	0	2	9	10	0	0	0	0	8	0	0	0	0	0	0	0	0	0
mv_DG	0	9	-	9	10	5	9	7	8	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
mv_NP	0	5	7	-	10	3	7	9	8	11	9	0	12	0	0	0	0	0	0	0	0	0	0	0	0
mv_HN	0	4	8	8	-	5	9	8	8	10	10	0	10	0	8	0	0	0	0	0	0	0	0	0	0
mv_HS	0	2	5	5	3	-	10	2	5	6	9	0	10	0	0	1	0	0	0	0	0	0	14	0	0
mv_SM	0	6	7	5	7	6	-	6	6	10	12	0	10	6	4	0	0	0	0	0	0	0	0	0	0
mv_JB	0	0	5	1	3	4	8	-	9	6	0	13	6	8	9	0	0	0	8	0	0	0	0	0	0
mv_PE	0	2	6	0	4	3	6	11	-	5	0	0	5	7	0	0	0	0	5	0	0	0	0	0	0
mv_FC	0	5	11	9	8	2	8	8	7	-	10	0	11	0	0	0	0	0	0	0	0	0	0	0	0
mv_TP	0	4	0	3	2	3	0	0	0	0	-	6	11	0	4	5	0	0	0	0	0	0	0	0	0
mv_CM	0	0	0	0	0	0	0	5	0	0	4	-	3	8	8	9	0	7	5	0	0	0	0	0	0
mv_DF	0	0	0	2	2	2	2	0	1	1	5	1	-	0	6	0	0	0	5	6	8	13	13	0	0
mv_TM	0	0	0	0	0	0	0	0	0	0	0	2	0	-	2	2	7	6	0	0	0	0	0	0	0
mv_DJ	0	0	0	0	2	4	2	1	0	0	0	0	0	4	-	3	5	0	0	0	4	3	1	0	0
mv_AL	0	0	0	0	0	1	0	0	0	0	3	1	0	2	3	-	2	2	0	0	3	1	0	0	0
mv_RG	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	4	-	1	0	6	0	8	9	0	0
mv_MB	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0	2	3	-	2	0	4	4	2	0	0
mv_GH	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	6	-	0	2	5	3	0	0
mv_HP	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	2	0	0	-	3	6	5	0	8
mv_BI	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	2	1	-	1	0	0	0	4
mv_DI	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	4	0	1	8	5	-	6	0	8
mv_FF	0	0	0	0	0	0	0	0	0	0	0	0	1	0	5	0	9	0	1	7	0	12	-	9	0
mv_GG	0	0	0	0	0	0	0	0	0	0	0	0	3	0	1	0	0	0	0	0	0	3	-	0	2
mv_TF	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	2	-

Valuation domain: [-19.00; +19.00]

Fig. 2.4: Symmetric part of graffiti07 digraph

Such a preordering of the movies may, for instance, be computed with the `computeRankingByChoosing()` method, where we iteratively extract *dominant kernels*

-remaining first choices- and *absorbent kernels* -remaining last choices- (see the tutorial on Computing Digraph Kernels). We operate therefore on the asymmetric ‘*better rated than*’, i.e. the *codual* ⁽³⁾ of the ‘*at least as well rated as*’ opinions (see Listing 2.9 Line 2).

Listing 2.9: Ranking by choosing the Grafitti movies

```

1 >>> from transitiveDigraphs import RankingByChoosingDigraph
2 >>> rbc = RankingByChoosingDigraph(g, CoDual=True)
3 >>> rbc.showRankingByChoosing()
4 Ranking by Choosing and Rejecting
5 1st First Choice ['mv_QS']
6 2nd First Choice ['mv_DG', 'mv_FC', 'mv_HN', 'mv_HS', 'mv_NP',
7                  'mv_PE', 'mv_RR', 'mv_SM']
8 3rd First Choice ['mv_CM', 'mv_JB', 'mv_TM']
9 4th First Choice ['mv_AL', 'mv_TP']
10 4th Last Choice ['mv_AL', 'mv_TP']
11 3rd Last Choice ['mv_GH', 'mv_MB', 'mv_RG']
12 2nd Last Choice ['mv_DF', 'mv_DJ', 'mv_FF', 'mv_GG']
13 1st Last Choice ['mv_BI', 'mv_DI', 'mv_HP', 'mv_TF']

```

In the tutorial on Computing Digraph Kernels, we thoroughly discuss the computation of *kernels* in bipolar-valued digraphs. Yet, we do not tackle there the problem of computing the corresponding bipolar-valued **kernel membership** characteristic vectors. This is the topic for the next pearl of bipolar-valued epistemic logic.

Back to [Content Table](#) (page 2)

3 Bipolar-valued kernel membership characteristic vectors

- [Kernel equation systems](#) (page 19)
- [Solving bipolar-valued kernel equation systems](#) (page 20)

³ A *kernel* in a digraph g is a *clique* in the dual digraph $-g$.

3.1 Kernel equation systems

Let $G(X, R)$ be a crisp irreflexive digraph defined on a finite set X of nodes and where R is the corresponding $\{-1, +1\}$ -valued adjacency matrix. Let Y be the $\{-1, +1\}$ -valued membership characteristic (row) vector of a choice in X . When Y satisfies the following equation system

$$Y \circ R = -Y ,$$

where for all x in X ,

$$(Y \circ R)(x) = \max_{y \in X, x \neq y} (\min(Y(x), R(x, y))) .$$

then Y characterises an **initial kernel** ([SCH-1985p]).

When transposing now the membership characteristic vector Y into a column vector Y^t , the following equation system

$$R \circ Y^t = -Y^t ,$$

makes Y^t similarly characterise a **terminal kernel**.

Let us verify this result on a tiny random digraph.

```

1  >>> from digraphs import *
2  >>> g = RandomDigraph(order=3, seed=1)
3  >>> g.showRelationTable()
4  * ---- Relation Table ----
5      R | 'a1'      'a2'      'a3'
6  -----|-----
7  'a1' |  -1   +1      -1
8  'a2' |  -1   -1      +1
9  'a3' |  +1   +1      -1
10 >>> g.showPreKernels()
11 *--- Computing preKernels ---*
12 Dominant preKernels :
13 ['a3']
14     independence : 1.0
15     dominance    : 1.0
16     absorbency   : -1.0
17     covering     : 1.000
18 Absorbent preKernels :
19 ['a2']
20     independence : 1.0
21     dominance    : -1.0
22     absorbency   : 1.0
23     covered     : 1.000

```

It is easy to verify that the characteristic vector $[-1, -1, +1]$ satisfies the initial kernel equation system; $a3$ gives an *initial* kernel. Similarly, the characteristic vector $[-1, +1, -1]$ verifies indeed the terminal kernel equation system and hence $a2$ gives a *terminal* kernel.

We succeeded now in generalizing kernel equation systems to genuine bipolar-valued digraphs ([BIS-2006_1p]). The constructive proof, found by *Marc Piriot*, is based on the following *fixpoint equation* that may be used for computing bipolar-valued kernel membership vectors,

$$T(Y) := -(Y \circ R) = Y,$$

3.2 Solving bipolar-valued kernel equation systems

John von Neumann showed indeed that, when a digraph $G(X, R)$ is **acyclic** with a **unique initial kernel** K characterised by its membership characteristics vector Y_k , then the following double bipolar-valued fixpoint equation

$$T^2(Y) := -(- (Y \circ R) \circ R) = Y.$$

will admit a stable high and a stable low fixpoint solution that converge both to Y_k ([SCH-1985p]).

Inspired by this crisp double fixpoint equation, we observed that for a given bipolar-valued digraph $G(X, R)$, each of its dominant or absorbent prekernels K_i in X determines an induced **partial graph** $G(X, R/K_i)$ which is *acyclic* and admits K_i as unique kernel (see [BIS-2006_2p]).

Following the *von Neumann* fixpoint algorithm, a similar bipolar-valued extended double fixpoint algorithm, applied to $G(X, R/K_i)$, allows to compute hence the associated bipolar-valued kernel characteristic vectors Y_i in polynomial complexity.

Algorithm

in : bipolar-valued digraph $G(X, R)$,

out : set $\{ Y_1, Y_2, \dots \}$ of bipolar-valued kernel membership characteristic vectors.

1. enumerate all initial and terminal crisp prekernels K, K_2, \dots in the given bipolar-valued digraph (see the tutorial on Computing Digraph Kernels);
2. for each crisp initial kernel K_i :
 - a. construct a partially determined subgraph $G(X, R/K_i)$ supporting exactly this unique initial kernel K_i ;
 - b. Use the double fixpoint equation T^2 with the partially determined adjacency matrix R/K_i for computing a stable low and a stable high fixpoint;
 - c. Determine the bipolar-valued K_i -membership characteristic vector Y_i with an epistemic disjunction of the previous low and high fixpoints;
3. repeat step (2) for each terminal kernel K_j by using the double fixpoint equation T^2 with the transpose of the adjacency matrix R/K_j .

Time for a practical illustration.

Listing 3.1: Random Bipolar-valued Outranking Digraph

```

1 >>> from outrankingDigraphs import *
2 >>> g = RandomBipolarOutrankingDigraph(Normalized=True,seed=5)
3 >>> print(g)
4 *----- Object instance description -----*
5 Instance class      : RandomBipolarOutrankingDigraph
6 Instance name      : rel_randomperftab
7 # Actions          : 7
8 # Criteria         : 7
9 Size              : 26
10 Determinateness (%) : 67.14
11 Valuation domain   : [-1.0;1.0]
12 Attributes         : ['name', 'actions', 'criteria', 'evaluation',
13                       'relation', 'valuationdomain', 'order',
14                       'gamma', 'notGamma']

```

The random outranking digraph g , we consider here in Listing 3.1 for illustration, models the pairwise outranking situations between seven decision alternatives evaluated on seven incommensurable performance criteria. We compute its corresponding bipolar-valued prekernels on the associated codual digraph gcd .

Listing 3.2: Strict Prekernels

```

1 >>> gcd = ~(-g) # strict outranking digraph
2 >>> gcd.showPreKernels()
3 *--- Computing prekernels ---*
4 Dominant prekernels :
5 ['a1', 'a4', 'a2']
6     independence : +0.000
7     dominance    : +0.070
8     absorbency   : -0.488
9     covering     : +0.667
10 Absorbent prekernels :
11 ['a7', 'a3']
12     independence : +0.000
13     dominance    : -0.744
14     absorbency   : +0.163
15     covered      : +0.800
16 *----- statistics -----
17 graph name: converse-dual_rel_randomperftab
18 number of solutions
19     dominant kernels : 1
20     absorbent kernels: 1
21 cardinality frequency distributions
22 cardinality        : [0, 1, 2, 3, 4, 5, 6, 7]
23 dominant kernel    : [0, 0, 0, 1, 0, 0, 0, 0]
24 absorbent kernel   : [0, 0, 1, 0, 0, 0, 0, 0]

```

(continues on next page)

25 Execution time : 0.00022 sec.

The codual outranking digraph, modelling a *strict outranking* relation, admits an initial prekernel $[a1, a2, a4]$ and a terminal one $[a3, a7]$ (see Listing 3.2 Line 5 and 11).

Let us compute the *initial* prekernel restricted adjacency table with the `domkernelrestrict()` method.

```
1 >>> k1Relation = gcd.domkernelrestrict(['a1','a2','a4'])
2 >>> gcd.showHTMLRelationTable(\
3 ...     actionsList=['a1','a2','a4','a3','a5','a6','a7'],\
4 ...     relation=k1Relation,\
5 ...     tableTitle='K1 restricted adjacency table')
```

K1 restricted adjacency table

r(x S y)	a1	a2	a4	a3	a5	a6	a7
a1	-	-0.23	-1.00	0.00	0.00	0.00	0.16
a2	-0.21	-	-0.21	0.21	0.44	0.05	0.49
a4	0.00	-0.21	-	0.21	0.00	0.07	0.58
a3	-0.28	-0.21	-0.74	-	0.00	0.00	0.00
a5	-0.26	-0.67	0.00	0.00	-	0.00	0.00
a6	-0.12	-0.49	-0.49	0.00	0.00	-	0.00
a7	-0.51	-0.49	-0.86	0.00	0.00	0.00	-

Valuation domain: [-1.00; +1.00]

Fig. 3.1: Initial kernel $[a1, a2, a4]$ restricted adjacency table

We first notice that this initial prekernel is indeed only *weakly independent*: The outranking situation between $a4$ and $a1$ appears *indeterminate*. The corresponding initial prekernel membership characteristic vector may be computed with the `computeKernelVector()` method.

Listing 3.3: Fixpoint iterations for initial prekernel $[a1, a2, a4]$

```
1 >>> gcd.computeKernelVector(['a1','a2','a4'],Initial=True,Comments=True)
2 --> Initial prekernel: {'a1', 'a2', 'a4'}
3 initial low vector : [-1.00, -1.00, -1.00, -1.00, -1.00, -1.00, -1.00]
4 initial high vector: [+1.00, +1.00, +1.00, +1.00, +1.00, +1.00, +1.00]
5 1st low vector      : [ 0.00, +0.21, -0.21,  0.00, -0.44, -0.07, -0.58]
6 1st high vector     : [+1.00, +1.00, +1.00, +1.00, +1.00, +1.00, +1.00]
7 2nd low vector      : [ 0.00, +0.21, -0.21,  0.00, -0.44, -0.07, -0.58]
8 2nd high vector     : [ 0.00, +0.21, -0.21, +0.21, -0.21, -0.05, -0.21]
9 3rd low vector      : [ 0.00, +0.21, -0.21, +0.21, -0.21, -0.07, -0.21]
10 3rd high vector     : [ 0.00, +0.21, -0.21, +0.21, -0.21, -0.05, -0.21]
```

(continues on next page)

(continued from previous page)

```
11 4th low vector      : [ 0.00, +0.21, -0.21, +0.21, -0.21, -0.07, -0.21]
12 4th high vector     : [ 0.00, +0.21, -0.21, +0.21, -0.21, -0.07, -0.21]
13 # iterations        : 4
14 low & high fusion   : [ 0.00, +0.21, -0.21, +0.21, -0.21, -0.07, -0.21]
15 Choice vector for initial prekernel: {'a1', 'a2', 'a4'}
16 a2: +0.21
17 a4: +0.21
18 a1:  0.00
19 a6: -0.07
20 a3: -0.21
21 a5: -0.21
22 a7: -0.21
```

We start the fixpoint computation with an empty set characterisation as first low vector and a complete set X characterising high vector. After each iteration, the low vector is set to the negation of the previous high vector and the high vector is set to the negation of the previous low vector.

A unique stable prekernel characteristic vector $Y1$ is here attained at the fourth iteration with positive members $a2$: +0.21 and $a4$: +0.21 (60.5% criteria significance majority); $a1$: 0.00 being an ambiguous potential member. Alternatives $a3$, $a5$, $a6$ and $a7$ are all negative members, i.e. positive **non members** of this outranking prekernel.

Let us now compute the restricted adjacency table for the outranked, i.e. the *terminal* prekernel $[a3, a7]$.

```
1 >>> k2Relation = gcd.abskernelrestrict(['a3','a7'])
2 >>> gcd.showHTMLRelationTable(\
3 ...     actionsList=['a3','a7','a1','a2','a4','a5','a6'],\
4 ...     relation=k2Relation,\
5 ...     tableTitle='K2 restricted adjacency table')
```

K2 restricted adjacency table

r(x S y)	a3	a7	a1	a2	a4	a5	a6
a3	-	0.00	-0.28	-0.21	-0.74	-0.40	-0.53
a7	-1.00	-	-0.51	-0.49	-0.86	-0.67	-0.63
a1	0.00	0.16	-	0.00	0.00	0.00	0.00
a2	0.21	0.49	0.00	-	0.00	0.00	0.00
a4	0.21	0.58	0.00	0.00	-	0.00	0.00
a5	0.00	0.16	0.00	0.00	0.00	-	0.00
a6	0.30	0.26	0.00	0.00	0.00	0.00	-

Valuation domain: [-1.00; +1.00]

Fig. 3.2: Terminal kernel $[a3, a7]$ restricted adjacency table

Again, we notice that this terminal prekernel is indeed only weakly independent. The

corresponding bipolar-valued characteristic vector $Y2$ may be computed as follows.

```

1 >>> gcd.computeKernelVector(['a3','a7'],Initial=False,Comments=True)
2 --> Terminal prekernel: {'a3', 'a7'}
3 initial low vector : [-1.00, -1.00, -1.00, -1.00, -1.00, -1.00, -1.00]
4 initial high vector : [+1.00, +1.00, +1.00, +1.00, +1.00, +1.00, +1.00]
5 1st low vector      : [-0.16, -0.49, 0.00, -0.58, -0.16, -0.30, +0.49]
6 1st high vector     : [+1.00, +1.00, +1.00, +1.00, +1.00, +1.00, +1.00]
7 2nd low vector      : [-0.16, -0.49, 0.00, -0.58, -0.16, -0.30, +0.49]
8 2nd high vector     : [-0.16, -0.49, 0.00, -0.49, -0.16, -0.26, +0.49]
9 3rd low vector      : [-0.16, -0.49, 0.00, -0.49, -0.16, -0.26, +0.49]
10 3rd high vector     : [-0.16, -0.49, 0.00, -0.49, -0.16, -0.26, +0.49]
11 # iterations       : 3
12 high & low fusion   : [-0.16, -0.49, 0.00, -0.49, -0.16, -0.26, +0.49]
13 Choice vector for terminal prekernel: {'a3', 'a7'}
14 a7: +0.49
15 a3: 0.00
16 a1: -0.16
17 a5: -0.16
18 a6: -0.26
19 a2: -0.49
20 a4: -0.49

```

A unique stable bipolar-valued high and low fixpoint is attained at the third iteration with $a7$ positively confirmed (about 75% criteria significance majority) as member of this terminal prekernel, whereas the membership of $a3$ in this prekernel appears indeterminate. All the remaining nodes have *negative* membership characteristic values and are hence positively excluded from this prekernel.

When we reconsider the graphviz drawing of this outranking digraph (see Fig. 52 in the tutorial on Computing Digraph Kernels),

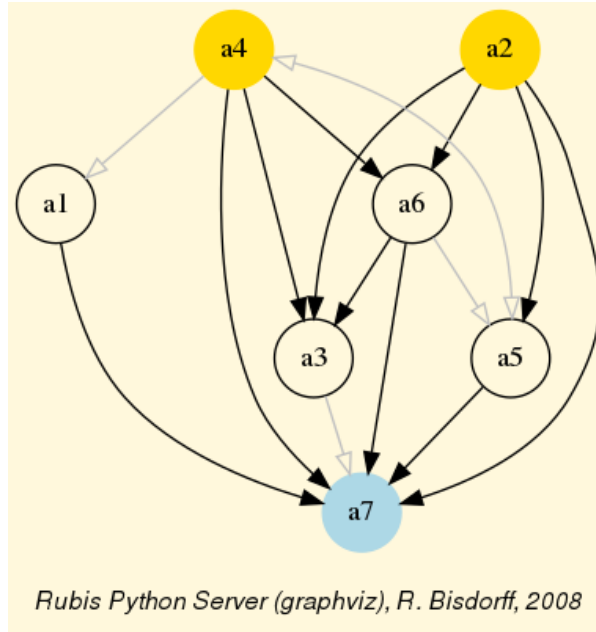


Fig. 3.3: The strict outranking digraph oriented by the positive members of its initial and terminal prekernels

it becomes obvious why alternative *a1* is **neither included nor excluded** from the initial prekernel. Same observation is applicable to alternative *a3* which can **neither be included nor excluded** from the terminal prekernel. It may even happen, in case of more indeterminate outranking situations, that no alternative is positively included or excluded from a weakly independent prekernel; the corresponding bipolar-valued membership characteristic vector being completely indeterminate (see for instance the tutorial on Computing a Best Choice Recommendation).

To illustrate finally why sometimes we need to operate an *epistemic disjunctive fusion* of **unequal** stable low and high membership characteristics vectors (see Step 2.c.), let us consider, for instance, the following crisp 7-cycle graph.

```

1  >>> g = CirculantDigraph(order=7,circulants=[-1,1])
2  >>> g
3  *----- Digraph instance description -----*
4  Instance class      : CirculantDigraph
5  Instance name      : c7
6  Digraph Order      : 7
7  Digraph Size       : 14
8  Valuation domain   : [-1.00;1.00]
9  Determinateness (%) : 100.00
10 Attributes         : ['name', 'order', 'circulants', 'actions',
11                       'valuationdomain', 'relation',
12                       'gamma', 'notGamma']

```

Digraph *c7* is a symmetric crisp digraph showing, among others, the maximal independent set {‘2’, ‘5’, ‘7’}, i.e. an initial as well as terminal kernel. We may compute the corresponding initial kernel characteristic vector.

```

1 >>> g.computeKernelVector(['2','5','7'],Initial=True,Comments=True)
2 --> Initial kernel: {'2', '5', '7'}
3 initial low vector   : [-1.0, -1.0, -1.0, -1.0, -1.0, -1.0, -1.0]
4 initial high vector  : [+1.0, +1.0, +1.0, +1.0, +1.0, +1.0, +1.0]
5 1 st low vector      : [-1.0,  0.0, -1.0, -1.0,  0.0, -1.0,  0.0]
6 1 st high vector     : [+1.0, +1.0, +1.0, +1.0, +1.0, +1.0, +1.0]
7 2 nd low vector      : [-1.0,  0.0, -1.0, -1.0,  0.0, -1.0,  0.0]
8 2 nd high vector     : [ 0.0, +1.0,  0.0,  0.0, +1.0,  0.0, +1.0]
9 stable low vector    : [-1.0,  0.0, -1.0, -1.0,  0.0, -1.0,  0.0]
10 stable high vector   : [ 0.0, +1.0,  0.0,  0.0, +1.0,  0.0, +1.0]
11 #iterations         : 3
12 low & high fusion    : [-1.0, +1.0, -1.0, -1.0, +1.0, -1.0, +1.0]
13 Choice vector for initial prekernel: {'2', '5', '7'}
14 7: +1.00
15 5: +1.00
16 2: +1.00
17 6: -1.00
18 4: -1.00
19 3: -1.00
20 1: -1.00

```

Notice that the stable low vector characterises the **negative membership** part, whereas, the stable high vector characterises the **positive membership** part (see Lines 9-10 above). The bipolar **disjunctive fusion** assembles eventually both stable parts into the correct prekernel characteristic vector (Line 12).

The adjacency matrix of a symmetric digraph staying *unchanged* by the transposition operator, the previous computations, when qualifying the same kernel as a *terminal* instance, will hence produce exactly the same result.

Note: It is worthwhile noticing again the essential computational role, the logical **indeterminate value 0.0** is playing in this double fixpoint algorithm. To implement such kind of algorithms without a logical **neutral term** would be like implementing numerical algorithms without a possible usage of the number 0. Infinitely many trivial *impossibility theorems* and *dubious logical results* come up.

Back to [Content Table](#) (page 2)

4 On confident outrankings with uncertain criteria significances

- *Modelling uncertain criteria significances* (page 27)
- *Bipolar-valued likelihood of ‘at least as good as’ situations* (page 28)
- *Confidence level of outranking situations* (page 31)

When modelling preferences following the outranking approach, the signs of the majority margins do sharply distribute validation and invalidation of pairwise outranking situations. How can we be confident in the resulting outranking digraph, when we acknowledge the usual imprecise knowledge of criteria significance weights coupled with small majority margins?

To answer this question, one usually requires *qualified* majority margins for confirming outranking situations. But how to choose such a qualifying majority level: two third, three fourth of the significances ?

In this tutorial we propose to link the qualifying significance majority with a required alpha%-confidence level. We model therefore the significance weights as random variables following more or less widespread distributions around an average significance value that corresponds to the given deterministic weight. As the bipolar-valued random credibility of an outranking statement hence results from the simple sum of positive or negative independent random variables, we may apply the Central Limit Theorem (CLT) for computing the *bipolar likelihood* that the expected majority margin will indeed be positive, respectively negative.

4.1 Modelling uncertain criteria significances

Let us consider the significance weights of a family F of m criteria to be **independent random variables** W_j , distributing the potential significance weights of each criterion $j = 1, \dots, m$ around a mean value $E(W_j)$ with variance $V(W_j)$.

Choosing a specific stochastic model of uncertainty is usually application specific. In the limited scope of this tutorial, we will illustrate the consequence of this design decision on the resulting outranking modelling with four slightly different models for taking into account the uncertainty with which we know the numerical significance weights: *uniform*, *triangular*, and two models of *Beta laws*, one more *widespread* and, the other, more *concentrated*.

When considering, for instance, that the potential range of a significance weight is distributed between 0 and two times its mean value, we obtain the following random variates:

1. A continuous **uniform** distribution on the range 0 to $2E(W_j)$. Thus $W_j \sim U(0, 2E(W_j))$ and $V(W_j) = 1/3(E(W_j))^2$;
2. A **symmetric beta** distribution with, for instance, parameters $\alpha = 2$ and $\beta = 2$. Thus, $W_i \sim \text{Beta}(2,2) * 2E(W_j)$ and $V(W_j) = 1/5(E(W_j))^2$.

3. A **symmetric triangular** distribution on the same range with mode $E(W_j)$. Thus $W_j \sim \text{Tr}(0, 2E(W_j), E(W_j))$ with $V(W_j) = 1/6(E(W_j))^2$;
4. A **narrower beta** distribution with for instance parameters $\alpha = 4$ and $\beta = 4$. Thus $W_j \sim \text{Beta}(4,4) * 2E(W_j)$, $V(W_j) = 1/9(E(W_j))^2$.

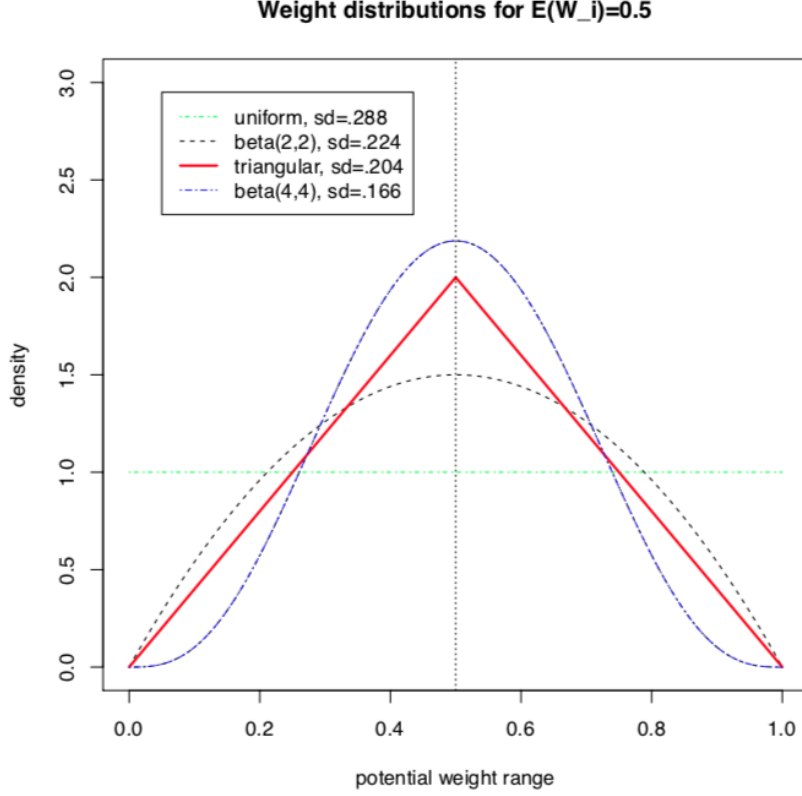


Fig. 4.1: Four models of uncertain significance weights

It is worthwhile noticing that these four uncertainty models all admit the same expected value, $E(W_j)$, however, with a respective variance which goes decreasing from $1/3$, to $1/9$ of the square of $E(W)$ (see Fig. 4.1).

4.2 Bipolar-valued likelihood of “at least as good as” situations

Let $A = \{x, y, z, \dots\}$ be a finite set of n potential decision actions, evaluated on $F = \{1, \dots, m\}$, a *finite* and *coherent* family of m performance criteria. On each criterion j in F , the decision actions are evaluated on a real performance scale $[0; M_j]$, supporting an upper-closed indifference threshold ind_j and a lower-closed preference threshold pr_j such that $0 \leq ind_j < pr_j \leq M_j$. The marginal performance of object x on criterion j is denoted x_j . Each criterion j is thus characterising a marginal double threshold order \geq_j on A (see Fig. 4.2):

$$r(x \geq_j y) = \begin{cases} +1 & \text{if } x_j - y_j \geq -ind_j, \\ -1 & \text{if } x_j - y_j \leq -pr_j, \\ 0 & \text{otherwise.} \end{cases}$$

Semantics of the marginal bipolar-valued characteristic function:

- +1 signifies x is performing at least as good as y on criterion j ,
- -1 signifies that x is not performing at least as good as y on criterion j ,
- 0 signifies that it is unclear whether, on criterion j , x is performing at least as good as y .

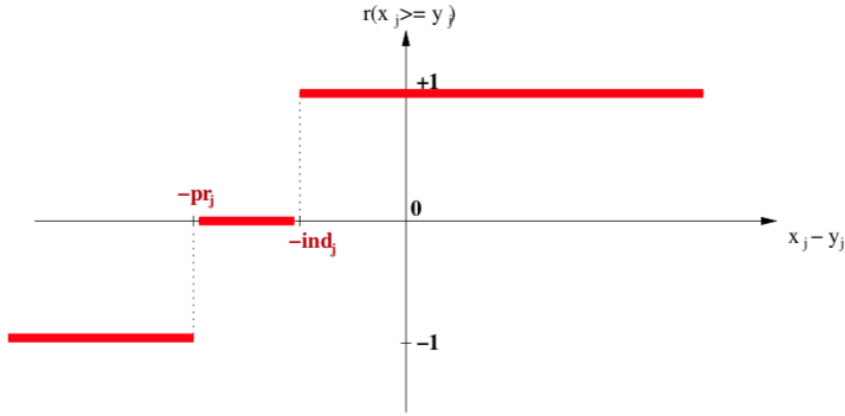


Fig. 4.2: Bipolar-valued outranking characteristic function

Each criterion j in F contributes the random significance W_j of his ‘at least as good as’ characteristic $r(x \geq_j y)$ to the global characteristic $\tilde{r}(x \geq y)$ in the following way:

$$\tilde{r}(x \geq y) = \sum_{j \in F} W_j \times r(x \geq_j y)$$

Thus, $\tilde{r}(x \geq y)$ becomes a simple sum of positive or negative independent random variables with known means and variances where $\tilde{r}(x \geq y) > 0$ signifies x is globally performing at least as good as y , $\tilde{r}(x \geq y) < 0$ signifies that x is not globally performing at least as good as y , and $\tilde{r}(x \geq y) = 0$ signifies that it is unclear whether x is globally performing at least as good as y .

From the *Central Limit Theorem* (CLT), we know that such a sum of random variables leads, with m getting large, to a Gaussian distribution Y with

$$E(Y) = \sum_{j \in F} (E(W_j) \times r(x \geq_j y)), \text{ and}$$

$$V(Y) = \sum_{j \in F} (V(W_j) \times |r(x \geq_j y)|).$$

And the **likelihood of validation**, respectively **invalidation** of an ‘*at least as good as*’ situation, denoted $lh(x \geq y)$, may hence be assessed by the probability $P(Y > 0) = 1.0 - P(Y \leq 0)$ that Y takes a positive, resp. $P(Y < 0)$ takes a negative value. In the bipolar-valued case here, we can judiciously make usage of the standard Gaussian **error function**, i.e. the bipolar $2P(Z) - 1.0$ version of the standard Gaussian $P(Z)$ probability distribution function:

$$lh(x \geq y) = -\text{erf}\left(\frac{1}{\sqrt{2}} \frac{-E(Y)}{\sqrt{V(Y)}}\right)$$

The range of the bipolar-valued $lh(x \geq y)$ hence becomes $[-1.0; +1.0]$, and $-lh(x \geq y) = lh(x \not\geq y)$, i.e. a **negative likelihood** represents the likelihood of the correspondent **negated** ‘*at least as good as*’ situation. A likelihood of $+1.0$ (resp. -1.0) means the corresponding preferential situation appears **certainly validated** (resp. **invalidated**).

Example

Let x and y be evaluated wrt 7 equisignificant criteria; Four criteria positively support that x is *at least as good performing* than y and three criteria support that x is *not at least as good performing* than y . Suppose $E(W_j) = w$ for $j = 1, \dots, 7$ and $W_j \sim \text{Tr}(0, 2w, w)$ for $j = 1, \dots, 7$. The expected value of the global ‘*at least as good as*’ characteristic value becomes: $E(\tilde{r}(x \geq y)) = 4w - 3w = w$ with a variance $V(\tilde{r}(x \geq y)) = 7\frac{1}{6}w^2$.

If $w = 1$, $E(\tilde{r}(x \geq y)) = 1$ and $sd(\tilde{r}(x \geq y)) = 1.08$. By the CLT, the bipolar likelihood of the *at least as good performing* situation becomes: $lh(x \geq y) = 0.66$, which corresponds to a global support of $(0.66 + 1.0)/2 = 83\%$ of the criteria significance weights.

A *Monte Carlo* simulation with 10 000 runs empirically confirms the effective convergence to a Gaussian (see Fig. 4.3 realised with *gretl*⁴).

⁴ The Gnu Regression, Econometrics and Time-series Library <http://gretl.sourceforge.net/> .

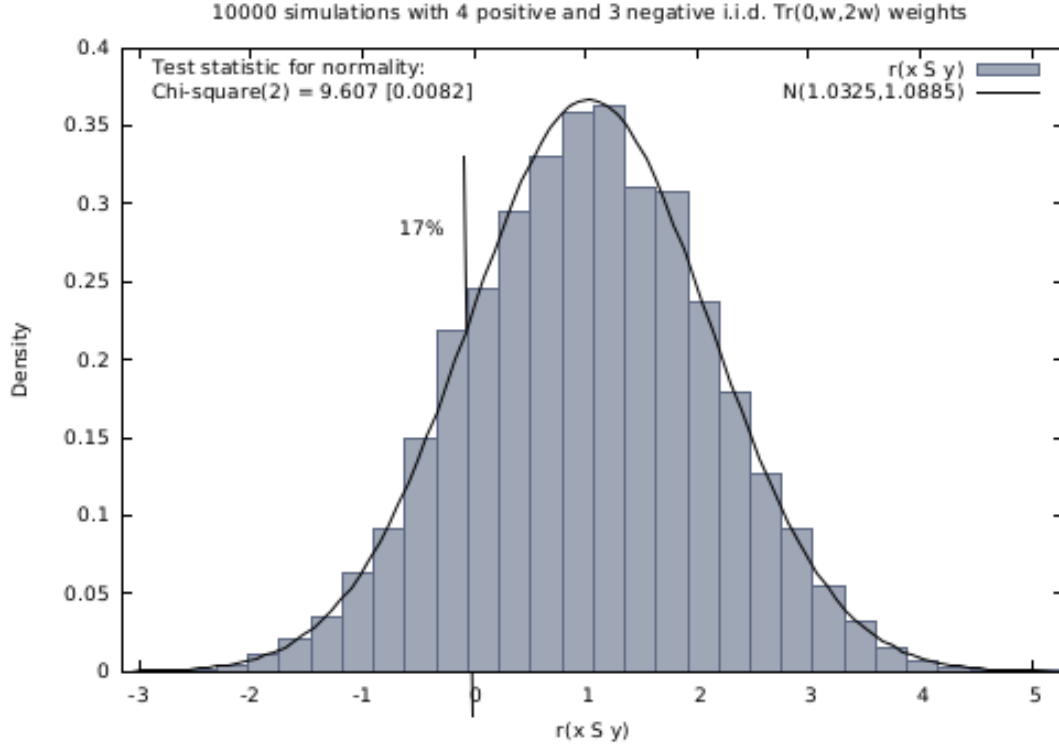


Fig. 4.3: Distribution of 10 000 random outranking characteristic values

Indeed, $\tilde{r}(x \geq y) \rightsquigarrow Y = \mathcal{N}(1.03, 1.089)$, with an empirical probability of observing a negative majority margin of about 17%.

4.3 Confidence level of outranking situations

Now, following the classical outranking approach (see [BIS-2013p]), we may say, from an epistemic perspective, that decision action x **outranks** decision action y at *confidence level α %*, when

1. an expected majority of criteria validates, at confidence level α % or higher, a global ‘*at least as good as*’ situation between x and y , and
2. no considerably less performing is observed on a discordant criterion.

Dually, decision action x **does not outrank** decision action y at confidence level α %, when

1. an expected majority of criteria at confidence level α % or higher, invalidates a global ‘*at least as good as*’ situation between x and y , and
2. no considerably better performing situation is observed on a concordant criterion.

Time for a coded example

Let us consider the following random performance tableau.


```

1 >>> from randomPerfTabs import RandomPerformanceTableau
2 >>> t = RandomPerformanceTableau(\
3 ...     numberOfActions=7,\
4 ...     numberOfCriteria=7,seed=100)
5
6 >>> t.showPerformanceTableau(Transposed=True)
7 *---- performance tableau ----*
8 criteria | weights | 'a1' 'a2' 'a3' 'a4' 'a5' 'a6' 'a7'
9 -----|-----
10 'g1' | 1 | 15.17 44.51 57.87 58.00 24.22 29.10 96.58
11 'g2' | 1 | 82.29 43.90 NA 35.84 29.12 34.79 62.22
12 'g3' | 1 | 44.23 19.10 27.73 41.46 22.41 21.52 56.90
13 'g4' | 1 | 46.37 16.22 21.53 51.16 77.01 39.35 32.06
14 'g5' | 1 | 47.67 14.81 79.70 67.48 NA 90.72 80.16
15 'g6' | 1 | 69.62 45.49 22.03 33.83 31.83 NA 48.80
16 'g7' | 1 | 82.88 41.66 12.82 21.92 75.74 15.45 6.05

```

For the corresponding confident outranking digraph, we require a confidence level of $\alpha = 90\%$. The `ConfidentBipolarOutrankingDigraph` class provides such a construction.

```

1 >>> from outrankingDigraphs import\
2 ...     ConfidentBipolarOutrankingDigraph
3
4 >>> g90 = ConfidentBipolarOutrankingDigraph(t,confidence=90)
5 >>> print(g90)
6 *----- Object instance description -----*
7 Instance class      : ConfidentBipolarOutrankingDigraph
8 Instance name       : rel_randomperftab_CLT
9 # Actions           : 7
10 # Criteria          : 7
11 Size               : 15
12 Uncertainty model   : triangular(a=0,b=2w)
13 Likelihood domain   : [-1.0;+1.0]
14 Confidence level    : 0.80 (90.0%)
15 Confident majority  : 0.14 (57.1%)
16 Determinateness (%) : 62.07
17 Valuation domain    : [-1.00;1.00]
18 Attributes          : ['name', 'bipolarConfidenceLevel',
19                        'distribution', 'betaParameter', 'actions',
20                        'order', 'valuationdomain', 'criteria',
21                        'evaluation', 'concordanceRelation',
22                        'vetos', 'negativeVetos',
23                        'largePerformanceDifferencesCount',
24                        'likelihoods', 'confidenceCutLevel',
25                        'relation', 'gamma', 'notGamma']

```

The resulting 90% confident expected outranking relation is shown below.

```

1 >>> g90.showRelationTable(LikelihoodDenotation=True)
2 * ---- Outranking Relation Table ----
3 r/(lh) | 'a1'      'a2'      'a3'      'a4'      'a5'      'a6'      'a7'
4 -----|-----
5 'a1' | +0.00   +0.71   +0.29   +0.29   +0.29   +0.29   +0.00
6       | ( - )   (+1.00) (+0.95) (+0.95) (+0.95) (+0.95) (+0.65)
7 'a2' | -0.71   +0.00   -0.29   +0.00   +0.00   +0.29   -0.57
8       | (-1.00) ( - )   (-0.95) (-0.65) (+0.73) (+0.95) (-1.00)
9 'a3' | -0.29   +0.29   +0.00   -0.29   +0.00   +0.00   -0.29
10      | (-0.95) (+0.95) ( - )   (-0.95) (-0.73) (-0.00) (-0.95)
11 'a4' | +0.00   +0.00   +0.57   +0.00   +0.29   +0.57   -0.43
12      | (-0.00) (+0.65) (+1.00) ( - )   (+0.95) (+1.00) (-0.99)
13 'a5' | -0.29   +0.00   +0.00   +0.00   +0.00   +0.29   -0.29
14      | (-0.95) (-0.00) (+0.73) (-0.00) ( - )   (+0.99) (-0.95)
15 'a6' | -0.29   +0.00   +0.00   -0.29   +0.00   +0.00   +0.00
16      | (-0.95) (-0.00) (+0.73) (-0.95) (+0.73) ( - )   (-0.00)
17 'a7' | +0.00   +0.71   +0.57   +0.43   +0.29   +0.00   +0.00
18      | (-0.65) (+1.00) (+1.00) (+0.99) (+0.95) (-0.00) ( - )
19 Valuation domain   : [-1.000; +1.000]
20 Uncertainty model   : triangular(a=2.0,b=2.0)
21 Likelihood domain   : [-1.0;+1.0]
22 Confidence level    : 0.80 (90.0%)
23 Confident majority   : 0.14 (57.1%)
24 Determinateness     : 0.24 (62.1%)

```

The (lh) figures, indicated in the table above, correspond to bipolar likelihoods and the required bipolar confidence level equals $(0.90+1.0)/2 = 0.80$ (see Line 22 above). Action ‘ $a1$ ’ thus confidently outranks all other actions, except ‘ $a7$ ’ where the actual likelihood (+0.65) is lower than the required one (0.80) and we furthermore observe a considerable counter-performance on criterion ‘ $g1$ ’.

Notice also the lack of confidence in the outranking situations we observe between action ‘ $a2$ ’ and actions ‘ $a4$ ’ and ‘ $a5$ ’. In the deterministic case we would have $r(a2 \geq a4) = -0.143$ and $r(a2 \geq a5) = +0.143$. All outranking situations with a characteristic value lower or equal to $\text{abs}(0.143)$, i.e. a majority support of $1.143/2 = 57.1\%$ and less, appear indeed to be *not confident* at level 90% (see Line 23 above).

We may draw the corresponding strict 90%-confident outranking digraph, oriented by its initial and terminal *strict* prekernels (see Fig. 4.4).

```

1 >>> gcd90 = ~ (-g90)
2 >>> gcd90.showPreKernels()
3 *--- Computing preKernels ---*
4 Dominant preKernels :
5 ['a1', 'a7']
6 independence : 0.0
7 dominance    : 0.2857
8 absorbency   : -0.7143

```

(continues on next page)

(continued from previous page)

```
9      covering      : 0.800
10 Absorbent preKernels :
11 ['a2', 'a5', 'a6']
12 independence      : 0.0
13 dominance         : -0.2857
14 absorbency        : 0.2857
15 covered          : 0.583
16 >>> gcd90.exportGraphViz(fileName='confidentOutranking',
17 ...     firstChoice=['a1', 'a7'],lastChoice=['a2', 'a5', 'a6'])
18
19 *---- exporting a dot file for GraphViz tools -----*
20 Exporting to confidentOutranking.dot
21 dot -Grankdir=BT -Tpng confidentOutranking.dot -o confidentOutranking.
  →png
```

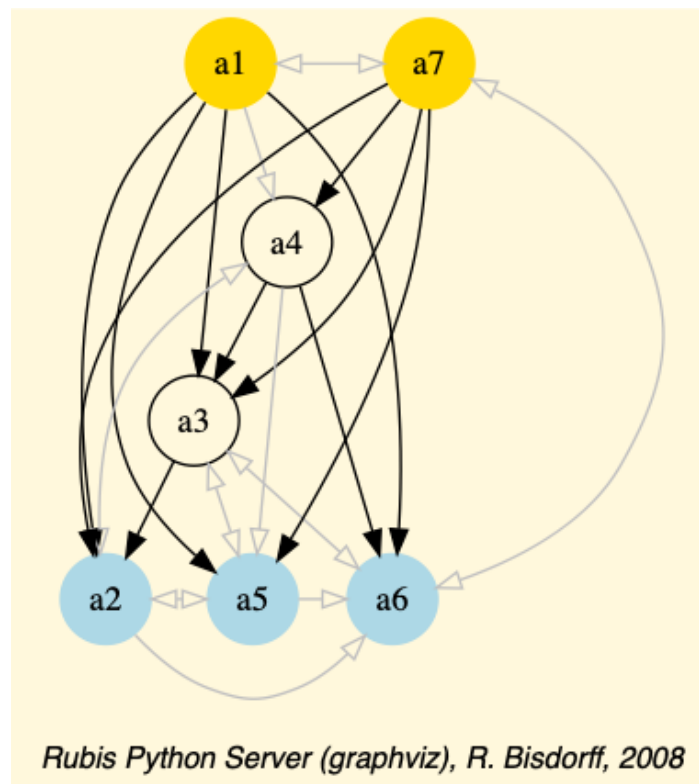


Fig. 4.4: Strict 90%-confident outranking digraph oriented by its pre-kernels

Now, what becomes this 90%-confident outranking digraph when we require a stronger confidence level of, say 99% ?

```
1 >>> g99 = ConfidentBipolarOutrankingDigraph(t,confidence=99)
2 >>> g99.showRelationTable()
3 * ---- Outranking Relation Table ----
4 r/(lh) | 'a1'      'a2'      'a3'      'a4'      'a5'      'a6'      'a7'
```

(continues on next page)

(continued from previous page)

```
5  -----|-----
6  'a1' | +0.00 +0.71 +0.00 +0.00 +0.00 +0.00 +0.00
7      | ( - ) (+1.00) (+0.95) (+0.95) (+0.95) (+0.95) (+0.65)
8  'a2' | -0.71 +0.00 +0.00 +0.00 +0.00 +0.00 +0.00 -0.57
9      | (-1.00) ( - ) (-0.95) (-0.65) (+0.73) (+0.95) (-1.00)
10 'a3' | +0.00 +0.00 +0.00 +0.00 +0.00 +0.00 +0.00
11      | (-0.95) (+0.95) ( - ) (-0.95) (-0.73) (-0.00) (-0.95)
12 'a4' | +0.00 +0.00 +0.57 +0.00 +0.00 +0.57 -0.43
13      | (-0.00) (+0.65) (+1.00) ( - ) (+0.95) (+1.00) (-0.99)
14 'a5' | +0.00 +0.00 +0.00 +0.00 +0.00 +0.29 +0.00
15      | (-0.95) (-0.00) (+0.73) (-0.00) ( - ) (+0.99) (-0.95)
16 'a6' | +0.00 +0.00 +0.00 +0.00 +0.00 +0.00 +0.00
17      | (-0.95) (-0.00) (+0.73) (-0.95) (+0.73) ( - ) (-0.00)
18 'a7' | +0.00 +0.71 +0.57 +0.43 +0.00 +0.00 +0.00
19      | (-0.65) (+1.00) (+1.00) (+0.99) (+0.95) (-0.00) ( - )
20 Valuation domain : [-1.000; +1.000]
21 Uncertainty model : triangular(a=2.0,b=2.0)
22 Likelihood domain : [-1.0;+1.0]
23 Confidence level : 0.98 (99.0%)
24 Confident majority : 0.29 (64.3%)
25 Determinateness : 0.13 (56.6%)
```

At 99% confidence, the minimal required significance majority support amounts to 64.3% (see Line 24 above). As a result, most outranking situations don't get anymore validated, like the outranking situations between action 'a1' and actions 'a3', 'a4', 'a5' and 'a6' (see Line 5 above). The overall epistemic determination of the digraph consequently drops from 62.1% to 56.6% (see Line 25).

Finally, what becomes the previous 90%-confident outranking digraph if the uncertainty concerning the criteria significance weights is modelled with a larger variance, like *uniform* variates (see Line 2 below).

```
1  >>> gu90 = ConfidentBipolarOutrankingDigraph(t,\
2  ...      confidence=90,distribution='uniform')
3
4  >>> gu90.showRelationTable()
5  * ---- Outranking Relation Table ----
6  r/(lh) | 'a1' 'a2' 'a3' 'a4' 'a5' 'a6' 'a7'
7  -----|-----
8  'a1' | +0.00 +0.71 +0.29 +0.29 +0.29 +0.29 +0.00
9      | ( - ) (+1.00) (+0.84) (+0.84) (+0.84) (+0.84) (+0.49)
10 'a2' | -0.71 +0.00 -0.29 +0.00 +0.00 +0.29 -0.57
11      | (-1.00) ( - ) (-0.84) (-0.49) (+0.56) (+0.84) (-1.00)
12 'a3' | -0.29 +0.29 +0.00 -0.29 +0.00 +0.00 -0.29
13      | (-0.84) (+0.84) ( - ) (-0.84) (-0.56) (-0.00) (-0.84)
14 'a4' | +0.00 +0.00 +0.57 +0.00 +0.29 +0.57 -0.43
15      | (-0.00) (+0.49) (+1.00) ( - ) (+0.84) (+1.00) (-0.95)
```

(continues on next page)

(continued from previous page)

16	'a5'		-0.29	+0.00	+0.00	+0.00	+0.00	+0.29	-0.29
17			(-0.84)	(-0.00)	(+0.56)	(-0.00)	(-)	(+0.92)	(-0.84)
18	'a6'		-0.29	+0.00	+0.00	-0.29	+0.00	+0.00	+0.00
19			(-0.84)	(-0.00)	(+0.56)	(-0.84)	(+0.56)	(-)	(-0.00)
20	'a7'		+0.00	+0.71	+0.57	+0.43	+0.29	+0.00	+0.00
21			(-0.49)	(+1.00)	(+1.00)	(+0.95)	(+0.84)	(-0.00)	(-)
22	Valuation domain : [-1.000; +1.000]								
23	Uncertainty model : uniform(a=2.0,b=2.0)								
24	Likelihood domain : [-1.0;+1.0]								
25	Confidence level : 0.80 (90.0%)								
26	Confident majority : 0.14 (57.1%)								
27	Determinateness : 0.24 (62.1%)								

Despite lower likelihood values (see the *g90* relation table above), we keep the same confident majority level of 57.1% (see Line 25 above) and, hence, also the same 90%-confident outranking digraph.

Note: For concluding, it is worthwhile noticing again that it is in fact the **neutral** value of our *bipolar-valued epistemic logic* that allows us to easily handle alpha% confidence or not of outranking situations when confronted with uncertain criteria significances. Remarkable furthermore is the usage, the standard **Gaussian error function** (erf) provides by delivering *signed likelihood values* immediately concerning either a *positive* relational statement, or when negative, its *negated* version.

[Back to *Content Table* \(page 2\)](#)

5 On stable outrankings with ordinal criteria significances

- *Cardinal or ordinal criteria significances* (page 37)
- *Qualifying the stability of outranking situations* (page 38)
- *Computing the stability denotation of outranking situations* (page 43)
- *Robust bipolar-valued outranking digraphs* (page 44)

5.1 Cardinal or ordinal criteria significances

The required cardinal significance weights of the performance criteria represent the *Achilles'* heel of the outranking approach. Rarely will indeed a decision maker be cognitively competent for suggesting precise decimal-valued criteria significance weights. More often, the decision problem will involve more or less equally important decision objectives with more or less equi-significant criteria. A random example of such a decision problem may be generated with the `Random3ObjectivesPerformanceTableau` class.

Listing 5.1: Random 3 Objectives Performance Tableau

```
1 >>> from randomPerfTabs import \
2 ...     Random3ObjectivesPerformanceTableau
3
4 >>> t = Random3ObjectivesPerformanceTableau(\
5 ...     numberOfActions=7,\
6 ...     numberOfCriteria=9,seed=102)
7
8 >>> t
9 *----- PerformanceTableau instance description -----*
10 Instance class      : Random3ObjectivesPerformanceTableau
11 Seed                : 102
12 Instance name       : random3ObjectivesPerfTab
13 # Actions           : 7
14 # Objectives         : 3
15 # Criteria          : 9
16 Attributes          : ['name', 'valueDigits', 'BigData', 'OrdinalScales',
17 →                    'missingDataProbability', 'negativeWeightProbability',
18 →                    'randomSeed', 'sumWeights', 'valuationPrecision',
19 →                    'commonScale', 'objectiveSupportingTypes', 'actions',
20 →                    'objectives', 'criteriaWeightMode', 'criteria',
21 →                    'evaluation', 'weightPreorder']
22 >>> t.showObjectives()
23 *----- show objectives -----"
24 Eco: Economical aspect
25   ec1 criterion of objective Eco 8
26   ec4 criterion of objective Eco 8
27   ec8 criterion of objective Eco 8
28   Total weight: 24.00 (3 criteria)
29 Soc: Societal aspect
30   so2 criterion of objective Soc 12
31   so7 criterion of objective Soc 12
32   Total weight: 24.00 (2 criteria)
33 Env: Environmental aspect
34   en3 criterion of objective Env 6
35   en5 criterion of objective Env 6
```

(continues on next page)

(continued from previous page)

```
36     en6 criterion of objective Env 6
37     en9 criterion of objective Env 6
38     Total weight: 24.00 (4 criteria)
```

In this example (see [Listing 5.1](#)), we face seven decision alternatives that are assessed with respect to three *equally important* decision objectives concerning: first, an *economical* aspect (Line 24) with a coalition of three performance criteria of significance weight 8, secondly, a *societal* aspect (Line 29) with a coalition of two performance criteria of significance weight 12, and thirdly, an *environmental* aspect (Line 33) with a coalition four performance criteria of significance weight 6.

The question we tackle is the following: How *dependent* on the actual values of the significance weights appears the corresponding bipolar-valued outranking digraph? In the previous section, we assumed that the criteria significance weights were random variables. Here, we shall assume that we know for sure only the preordering of the significance weights. In our example we see indeed three increasing weight equivalence classes ([Listing 5.2](#)).

Listing 5.2: Significance weights preorder

```
1 >>> t.showWeightPreorder()
2 ['en3', 'en5', 'en6', 'en9'] (6) <
3 ['ec1', 'ec4', 'ec8'] (8) <
4 ['so2', 'so7'] (12)
```

How stable appear now the outranking situations when assuming only ordinal significance weights?

5.2 Qualifying the stability of outranking situations

Let us construct the normalized bipolar-valued outranking digraph corresponding with the previous 3 Objectives performance tableau *t*.

Listing 5.3: Example Bipolar Outranking Digraph

```
1 >>> from outrankingDigraphs import BipolarOutrankingDigraph
2 >>> g = BipolarOutrankingDigraph(t, Normalized=True)
3 >>> g.showRelationTable()
4 * ---- Relation Table ----
5 r(>=) | 'p1'  'p2'  'p3'  'p4'  'p5'  'p6'  'p7'
6 -----|-----
7 'p1' | +1.00 -0.42 +0.00 -0.69 +0.39 +0.11 -0.06
8 'p2' | +0.58 +1.00 +0.83 +0.00 +0.58 +0.58 +0.58
9 'p3' | +0.25 -0.33 +1.00 +0.00 +0.50 +1.00 +0.25
10 'p4' | +0.78 +0.00 +0.61 +1.00 +1.00 +1.00 +0.67
11 'p5' | -0.11 -0.50 -0.25 -0.89 +1.00 +0.11 -0.14
12 'p6' | +0.22 -0.42 +0.00 -1.00 +0.17 +1.00 -0.11
```

(continues on next page)

13

'p7'		+0.22	-0.50	+0.17	-0.06	+0.78	+0.42	+1.00
------	--	-------	-------	-------	-------	-------	-------	-------

We notice on the principal diagonal, the *certainly validated* reflexive terms +1.00 (see Listing 5.3 Lines 7-13). Now, we know for sure that *unanimous* outranking situations are completely independent of the significance weights. Similarly, all outranking situations that are supported by a *majority* significance in *each* coalition of equi-significant criteria are also in fact independent of the actual importance we attach to each individual criteria coalition. But we are also able to test (see [BIS-2014p]) if an outranking situation is independent of all the potential significance weights that respect the given *preordering* of the weights. Mind that there are, for sure, always outranking situations that are indeed *dependent* on the very values we allocate to the criteria significances.

Such a stability denotation of outranking situations is readily available with the common `showRelationTable()` method.

Listing 5.4: Relation Table with Stability Denotation

```

1 >>> g.showRelationTable(StabilityDenotation=True)
2 * ---- Relation Table ----
3 r/(stab) | 'p1' 'p2' 'p3' 'p4' 'p5' 'p6' 'p7'
4 -----|-----
5 'p1' | +1.00 -0.42 +0.00 -0.69 +0.39 +0.11 -0.06
6 | (+4) (-2) (+0) (-3) (+2) (+2) (-1)
7 'p2' | +0.58 +1.00 +0.83 0.00 +0.58 +0.58 +0.58
8 | (+2) (+4) (+3) (+2) (+2) (+2) (+2)
9 'p3' | +0.25 -0.33 +1.00 0.00 +0.50 +1.00 +0.25
10 | (+2) (-2) (+4) (0) (+2) (+2) (+1)
11 'p4' | +0.78 0.00 +0.61 +1.00 +1.00 +1.00 +0.67
12 | (+3) (-1) (+3) (+4) (+4) (+4) (+2)
13 'p5' | -0.11 -0.50 -0.25 -0.89 +1.00 +0.11 -0.14
14 | (-2) (-2) (-2) (-3) (+4) (+2) (-2)
15 'p6' | +0.22 -0.42 0.00 -1.00 +0.17 +1.00 -0.11
16 | (+2) (-2) (+1) (-2) (+2) (+4) (-2)
17 'p7' | +0.22 -0.50 +0.17 -0.06 +0.78 +0.42 +1.00
18 | (+2) (-2) (+1) (-1) (+3) (+2) (+4)

```

We may thus distinguish the following bipolar-valued stability levels:

- **+4 | -4** : *unanimous* outranking | outranked situation. The pairwise trivial reflexive outrankings, for instance, all show this stability level;
- **+3 | -3** : *validated* outranking | outranked situation in *each* coalition of equi-significant criteria. This is, for instance, the case for the outranking situation observed between alternatives *p1* and *p4* (see Listing 5.4 Lines 6 and 12);
- **+2 | -2** : outranking | outranked situation *validated* with *all* potential significance weights that are *compatible* with the given significance *preorder* (see Listing 5.2. This is case for the comparison of alternatives *p1* and *p2* (see Listing 5.4 Lines 6 and 8);

- **+1 | -1** : *validated* outranking | outranked situation with the given significance weights, a situation we may observe between alternatives $p3$ and $p7$ (see Listing 5.4 Lines 10 and 16);
- **0** : *indeterminate* relational situation, like the one between alternatives $p1$ and $p3$ (see Listing 5.4 Lines 6 and 10).

It is worthwhile noticing that, in the one limit case where all performance criteria appear equi-significant, i.e. there is given a single equivalence class containing all the performance criteria, we may only distinguish stability levels +4 and +3 (rep. -4 and -3). Furthermore, when in such a case an outranking (resp. outranked) situation is validated at level +3 (resp. -3), no potential preordering of the criteria significances exists that could qualify the same situation as outranked (resp. outranking) at level -2 (resp. +2).

In the other limit case, when all performance criteria admit different significances, i.e. the significance weights may be linearly ordered, no stability level +3 or -3 may be observed.

As mentioned above, all *reflexive* comparisons confirm an unanimous outranking situation: all decision alternatives are indeed trivially *as well performing as* themselves. But there appear also two non reflexive unanimous outranking situations: when comparing, for instance, alternative $p4$ with alternatives $p5$ and $p6$ (see Listing 5.4 Lines 14 and 16).

Let us inspect the details of how alternatives $p4$ and $p5$ compare.

Listing 5.5: Comparing Decision Alternatives $a4$ and $a5$

```

1 >>> g.showPairwiseComparison('p4','p5')
2 *----- pairwise comparison -----*
3 Comparing actions : (p4, p5)
4 crit. wght. g(x) g(y) diff | ind pref r() |
5 ec1 8.00 85.19 46.75 +38.44 | 5.00 10.00 +8.00 |
6 ec4 8.00 72.26 8.96 +63.30 | 5.00 10.00 +8.00 |
7 ec8 8.00 44.62 35.91 +8.71 | 5.00 10.00 +8.00 |
8 en3 6.00 80.81 31.05 +49.76 | 5.00 10.00 +6.00 |
9 en5 6.00 49.69 29.52 +20.17 | 5.00 10.00 +6.00 |
10 en6 6.00 66.21 31.22 +34.99 | 5.00 10.00 +6.00 |
11 en9 6.00 50.92 9.83 +41.09 | 5.00 10.00 +6.00 |
12 so2 12.00 49.05 12.36 +36.69 | 5.00 10.00 +12.00 |
13 so7 12.00 55.57 44.92 +10.65 | 5.00 10.00 +12.00 |
14 Valuation in range: -72.00 to +72.00; global concordance: +72.00

```

Alternative $p4$ is indeed performing unanimously *at least as well as* alternative $p5$: $r(p4 \text{ outranks } p5) = +1.00$ (see Listing 5.4 Line 11).

The converse comparison does not, however, deliver such an unanimous *outranked* situation. This comparison only qualifies at stability level -3 (see Listing 5.4 Line 13 $r(p5 \text{ outranks } p4) = 0.89$).

Listing 5.6: Comparing Decision Alternatives $p5$ and $p4$

```

1 >>> g.showPairwiseComparison('p5','p4')
2 *----- pairwise comparison -----*

```

(continues on next page)

(continued from previous page)

```
3 Comparing actions : (p5, p4)
4 crit. wght. g(x) g(y) diff | ind pref r() |
5 ec1 8.00 46.75 85.19 -38.44 | 5.00 10.00 -8.00 |
6 ec4 8.00 8.96 72.26 -63.30 | 5.00 10.00 -8.00 |
7 ec8 8.00 35.91 44.62 -8.71 | 5.00 10.00 +0.00 |
8 en3 6.00 31.05 80.81 -49.76 | 5.00 10.00 -6.00 |
9 en5 6.00 29.52 49.69 -20.17 | 5.00 10.00 -6.00 |
10 en6 6.00 31.22 66.21 -34.99 | 5.00 10.00 -6.00 |
11 en9 6.00 9.83 50.92 -41.09 | 5.00 10.00 -6.00 |
12 so2 12.00 12.36 49.05 -36.69 | 5.00 10.00 -12.00 |
13 so7 12.00 44.92 55.57 -10.65 | 5.00 10.00 -12.00 |
14 Valuation in range: -72.00 to +72.00; global concordance: -64.00
```

Indeed, on criterion *ec8* we observe a small negative performance difference of -8.71 (see Listing 5.6 Line 7) which is effectively below the supposed *preference discrimination threshold* of 10.00. Yet, the outranked situation is supported by a majority of criteria in each decision objective. Hence, the reported preferential situation is completely independent of any chosen significance weights.

Let us now consider a comparison, like the one between alternatives *p2* and *p1*, that is only qualified at stability level +2, resp. -2.

Listing 5.7: Comparing Decision Alternatives *p2* and *p1*

```
1 >>> g.showPairwiseOutrankings('p2', 'p1')
2 *----- pairwise comparison -----*
3 Comparing actions : (p2, p1)
4 crit. wght. g(x) g(y) diff | ind pref r() |
5 ec1 8.00 89.77 38.11 +51.66 | 5.00 10.00 +8.00 |
6 ec4 8.00 86.00 22.65 +63.35 | 5.00 10.00 +8.00 |
7 ec8 8.00 89.43 77.02 +12.41 | 5.00 10.00 +8.00 |
8 en3 6.00 20.79 58.16 -37.37 | 5.00 10.00 -6.00 |
9 en5 6.00 23.83 31.40 -7.57 | 5.00 10.00 +0.00 |
10 en6 6.00 18.66 11.41 +7.25 | 5.00 10.00 +6.00 |
11 en9 6.00 26.65 44.37 -17.72 | 5.00 10.00 -6.00 |
12 so2 12.00 89.12 22.43 +66.69 | 5.00 10.00 +12.00 |
13 so7 12.00 84.73 28.41 +56.32 | 5.00 10.00 +12.00 |
14 Valuation in range: -72.00 to +72.00; global concordance: +42.00
15 *----- pairwise comparison -----*
16 Comparing actions : (p1, p2)
17 crit. wght. g(x) g(y) diff | ind pref r() |
18 ec1 8.00 38.11 89.77 -51.66 | 5.00 10.00 -8.00 |
19 ec4 8.00 22.65 86.00 -63.35 | 5.00 10.00 -8.00 |
20 ec8 8.00 77.02 89.43 -12.41 | 5.00 10.00 -8.00 |
21 en3 6.00 58.16 20.79 +37.37 | 5.00 10.00 +6.00 |
22 en5 6.00 31.40 23.83 +7.57 | 5.00 10.00 +6.00 |
23 en6 6.00 11.41 18.66 -7.25 | 5.00 10.00 +0.00 |
```

(continues on next page)

(continued from previous page)

```
24 en9  6.00  44.37  26.65  +17.72 | 5.00  10.00  +6.00      |
25 so2 12.00  22.43  89.12  -66.69 | 5.00  10.00  -12.00     |
26 so7 12.00  28.41  84.73  -56.32 | 5.00  10.00  -12.00     |
27 Valuation in range: -72.00 to +72.00; global concordance: -30.00
```

In both comparisons, the performances observed with respect to the environmental decision objective are not validating with a significant majority the otherwise unanimous outranking, resp. outranked situations. Hence, the stability of the reported preferential situations is in fact dependent on choosing significance weights that are compatible with the given significance weights preorder (see *Significance weights preorder* (page 38)).

Let us finally inspect a comparison that is only qualified at stability level +1, like the one between alternatives $p7$ and $p3$ (see Listing 5.8).

Listing 5.8: Comparing Decision Alternatives $p7$ and $p3$

```
1 >>> g.showPairwiseOutrankings('p7','p3')
2 *----- pairwise comparison -----*
3 Comparing actions : (p7, p3)
4 crit. wght.  g(x)  g(y)    diff | ind  pref  r()      |
5 ec1   8.00  15.33  80.19  -64.86 | 5.00  10.00  -8.00     |
6 ec4   8.00  36.31  68.70  -32.39 | 5.00  10.00  -8.00     |
7 ec8   8.00  38.31  91.94  -53.63 | 5.00  10.00  -8.00     |
8 en3   6.00  30.70  46.78  -16.08 | 5.00  10.00  -6.00     |
9 en5   6.00  35.52  27.25   +8.27 | 5.00  10.00  +6.00     |
10 en6   6.00  69.71   1.65 +68.06 | 5.00  10.00  +6.00     |
11 en9   6.00  13.10  14.85   -1.75 | 5.00  10.00  +6.00     |
12 so2  12.00  68.06  58.85   +9.21 | 5.00  10.00 +12.00    |
13 so7  12.00  58.45  15.49 +42.96 | 5.00  10.00 +12.00    |
14 Valuation in range: -72.00 to +72.00; global concordance: +12.00
15 *----- pairwise comparison -----*
16 Comparing actions : (p3, p7)
17 crit. wght.  g(x)  g(y)    diff | ind  pref  r()      |
18 ec1   8.00  80.19  15.33 +64.86 | 5.00  10.00  +8.00     |
19 ec4   8.00  68.70  36.31 +32.39 | 5.00  10.00  +8.00     |
20 ec8   8.00  91.94  38.31 +53.63 | 5.00  10.00  +8.00     |
21 en3   6.00  46.78  30.70 +16.08 | 5.00  10.00  +6.00     |
22 en5   6.00  27.25  35.52  -8.27 | 5.00  10.00  +0.00     |
23 en6   6.00   1.65  69.71 -68.06 | 5.00  10.00  -6.00     |
24 en9   6.00  14.85  13.10  +1.75 | 5.00  10.00  +6.00     |
25 so2  12.00  58.85  68.06  -9.21 | 5.00  10.00  +0.00     |
26 so7  12.00  15.49  58.45 -42.96 | 5.00  10.00 -12.00     |
27 Valuation in range: -72.00 to +72.00; global concordance: +18.00
```

In both cases, choosing significances that are just compatible with the given weights preorder will not always result in positively validated outranking situations.

5.3 Computing the stability denotation of outranking situations

Stability levels 4 and 3 are easy to detect, the case given. Detecting a stability level 2 is far less obvious. Now, it is precisely again the bipolar-valued epistemic characteristic domain that will give us a way to implement an effective test for stability level +2 and -2 (see [BIS-2004_1p], [BIS-2004_2p]).

Let us consider the significance equivalence classes we observe in the given weights pre-order. Here we observe three classes: 6, 8, and 12, in increasing order (see Listing 5.2). In the pairwise comparisons shown above these equivalence classes may appear positively or negatively, besides the indeterminate significance of value 0. We thus get the following ordered bipolar list of significance weights:

$$W = [-12, -8, -6, 0, 6, 8, 12].$$

In all the pairwise marginal comparisons shown in the previous Section, we may observe that each one of the nine criteria assigns one precise item out of this list W . Let us denote $q[i]$ the number of criteria assigning item $W[i]$, and $Q[i]$ the cumulative sums of these $q[i]$ counts, where i is an index in the range of the length of list W .

In the comparison of alternatives $a2$ and $a1$, for instance (see Listing 5.7), we observe the following counts:

$W[i]$	-12	-8	-6	0	6	8	12
$q[i]$	0	0	2	1	1	3	2
$Q[i]$	0	0	2	3	4	7	9

Let us denote $-q$ and $-Q$ the reversed versions of the q and the Q lists. We thus obtain the following result.

$W[i]$	-12	-8	-6	0	6	8	12
$-q[i]$	2	3	1	1	2	0	0
$-Q[i]$	2	5	6	7	9	9	9

Now, a pairwise outranking situation will be qualified at stability level +2, i.e. positively validated with any significance weights that are compatible with the given weights pre-order, when for all i , we observe $Q[i] \leq -Q[i]$ and there exists one i such that $Q[i] < -Q[i]$. Similarly, a pairwise outranked situation will be qualified at stability level -2, when for all i , we observe $Q[i] \geq -Q[i]$ and there exists one i such that $Q[i] > -Q[i]$ (see [BIS-2004_2p]).

We may verify, for instance, that the outranking situation observed between $a2$ and $a1$ does indeed verify this *first order distributional dominance* condition.

$W[i]$	-12	-8	-6	0	6	8	12
$Q[i]$	0	0	2	3	4	7	9
$-Q[i]$	2	5	6	7	9	9	9

Notice that outranking situations qualified at stability levels 4 and 3, evidently also verify

the stability level 2 test above. The outranking situation between alternatives a_7 and a_3 does not, however, verify this test (see Listing 5.8).

$W[i]$	-12	-8	-6	0	6	8	12
$q[i]$	0	3	1	0	3	0	2
$Q[i]$	0	3	4	4	7	7	9
$-Q[i]$	2	2	5	5	6	9	9

This time, *not* all the $Q[i]$ are *lower or equal* than the corresponding $-Q[i]$ terms. Hence the outranking situation between a_7 and a_3 is not positively validated with all potential significance weights that are compatible with the given weights preorder.

Using this stability denotation, we may, hence, define the following **robust** version of a bipolar-valued outranking digraph.

5.4 Robust bipolar-valued outranking digraphs

We say that decision alternative x **robustly outranks** decision alternative y when

- x positively outranks y at stability level *higher or equal to 2* and we may not observe any *considerable counter-performance* of x on a discordant criterion.

Dually, we say that decision alternative x **does not robustly outrank** decision alternative y when

- x negatively outranks y at stability level *lower or equal to -2* and we may not observe any *considerable better performance* of x on a discordant criterion.

The corresponding *robust* outranking digraph may be computed with the RobustOutrankingDigraph class as follows.

Listing 5.9: Robust outranking digraph

```

1  >>> from outrankingDigraphs import RobustOutrankingDigraph
2  >>> rg = RobustOutrankingDigraph(t) # same t as before
3  >>> rg
4  *----- Object instance description -----*
5  Instance class      : RobustOutrankingDigraph
6  Instance name      : robust_random3ObjectivesPerfTab
7  # Actions          : 7
8  # Criteria         : 9
9  Size               : 22
10 Determinateness (%) : 68.45
11 Valuation domain   : [-1.00;1.00]
12 Attributes         : ['name', 'methodData', 'actions', 'order',
13                       'criteria', 'evaluation', 'vetos',
14                       'valuationdomain', 'cardinalRelation',
15                       'ordinalRelation', 'equisignificantRelation',
16                       'unanimousRelation', 'relation',

```

(continues on next page)

(continued from previous page)

```

17                                     'gamma', 'notGamma']
18 >>> rg.showRelationTable(StabilityDenotation=True)
19 * ---- Relation Table ----
20 r/(stab) | 'p1'  'p2'  'p3'  'p4'  'p5'  'p6'  'p7'
21 -----|-----
22 'p1'  | +1.00 -0.42 +0.00 -0.69 +0.39 +0.11 +0.00
23       | (+4)  (-2)  (+0)  (-3)  (+2)  (+2)  (-1)
24 'p2'  | +0.58 +1.00 +0.83 +0.00 +0.58 +0.58 +0.58
25       | (+2)  (+4)  (+3)  (+2)  (+2)  (+2)  (+2)
26 'p3'  | +0.25 -0.33 +1.00 +0.00 +0.50 +1.00 +0.00
27       | (+2)  (-2)  (+4)  (+0)  (+2)  (+2)  (+1)
28 'p4'  | +0.78 +0.00 +0.61 +1.00 +1.00 +1.00 +0.67
29       | (+3)  (-1)  (+3)  (+4)  (+4)  (+4)  (+2)
30 'p5'  | -0.11 -0.50 -0.25 -0.89 +1.00 +0.11 -0.14
31       | (-2)  (-2)  (-2)  (-3)  (+4)  (+2)  (-2)
32 'p6'  | +0.22 -0.42 +0.00 -1.00 +0.17 +1.00 -0.11
33       | (+2)  (-2)  (+1)  (-2)  (+2)  (+4)  (-2)
34 'p7'  | +0.22 -0.50 +0.00 +0.00 +0.78 +0.42 +1.00
35       | (+2)  (-2)  (+1)  (-1)  (+3)  (+2)  (+4)

```

We may notice that all outranking situations, qualified at stability level +1 or -1, are now put to an *indeterminate* status. In the example here, we actually drop three positive outrankings: between $p3$ and $p7$, between $p7$ and $p3$, and between $p6$ and $p3$, where the last situation is actually already put to doubt by a veto situation (see Listing 5.9 Lines 22-35). We drop as well three negative outrankings: between $p1$ and $p7$, between $p4$ and $p2$, and between $p7$ and $p4$ (see Listing 5.9 Lines 22-35).

Notice by the way that outranking (resp. outranked) situations, although qualified at level +2 or +3 (resp. -2 or -3) may nevertheless be put to doubt by considerable performance differences. We may observe such an outranking situation when comparing, for instance, alternatives $p2$ and $p4$ (see Listing 5.9 Lines 24-25).

Listing 5.10: Comparing alternatives $p2$ and $p4$

```

1 >>> rg.showPairwiseComparison('p2', 'p4')
2 *----- pairwise comparison -----*
3 Comparing actions : (p2, p4)
4 crit. wght. g(x) g(y) diff | ind pref r() | v  □
5 -----|-----
6 ec1  8.00  89.77  85.19  +4.58 | 5.00 10.00 +8.00 |
7 ec4  8.00  86.00  72.26  +13.74 | 5.00 10.00 +8.00 |
8 ec8  8.00  89.43  44.62  +44.81 | 5.00 10.00 +8.00 |
9 en3  6.00  20.79  80.81  -60.02 | 5.00 10.00 -6.00 | 60.00 -1.
10 en5  6.00  23.83  49.69  -25.86 | 5.00 10.00 -6.00 |

```

(continues on next page)

(continued from previous page)

11	en6	6.00	18.66	66.21	-47.55		5.00	10.00	-6.00	
12	en9	6.00	26.65	50.92	-24.27		5.00	10.00	-6.00	
13	so2	12.00	89.12	49.05	+40.07		5.00	10.00	+12.00	
14	so7	12.00	84.73	55.57	+29.16		5.00	10.00	+12.00	
15	Valuation in range: -72.00 to +72.00; global concordance: +24.00									

Despite being robust, the apparent positive outranking situation between alternatives $p2$ and $p4$ is indeed put to doubt by a considerable counter-performance (-60.02) of $p2$ on criterion $en3$, a negative difference which exceeds slightly the assumed veto discrimination threshold $v = 60.00$ (see Listing 5.10 Line 9).

We may finally compare in Fig. 5.1 the *standard* and the *robust* version of the corresponding strict outranking digraphs, both oriented by their respective identical initial and terminal prekernels.

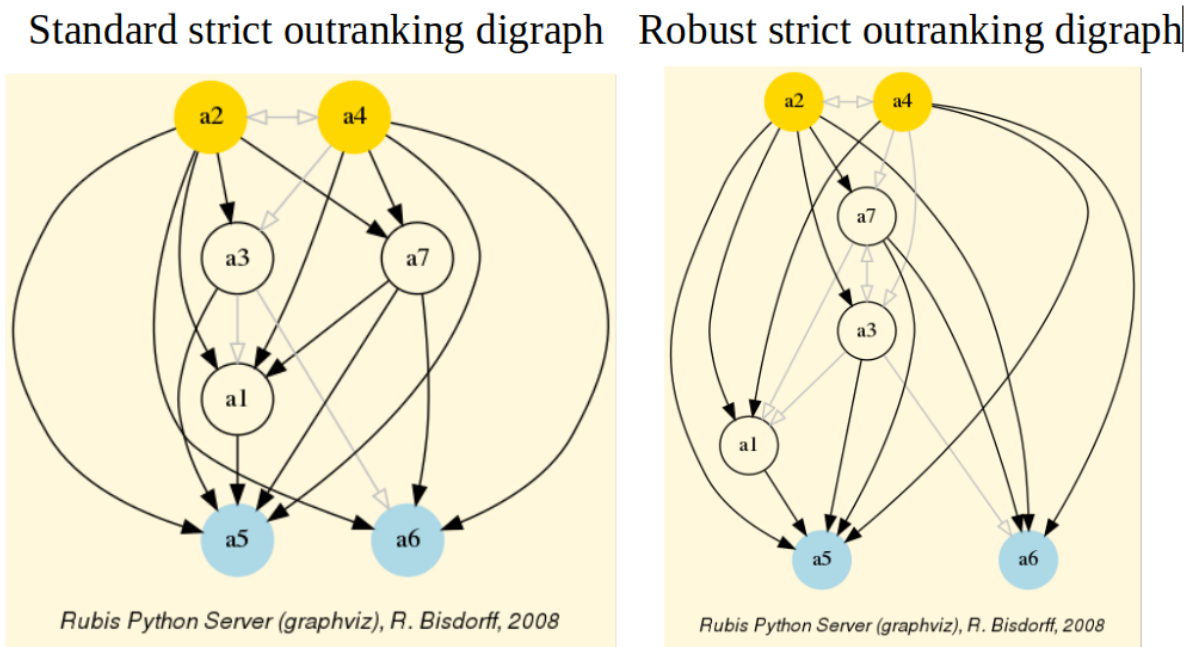


Fig. 5.1: Standard versus robust strict outranking digraphs oriented by their initial and terminal prekernels

The robust version drops two strict outranking situations: between $p4$ and $p7$ and between $p7$ and $p1$. The remaining 14 strict outranking (resp. outranked) situations are now all verified at a stability level of +2 and more (resp. -2 and less). They are, hence, only depending on potential significance weights that must respect the given significance preorder (see Listing 5.2).

To appreciate the apparent orientation of the standard and robust strict outranking digraphs shown in Fig. 5.1, let us have a final heat map view on the underlying performance tableau ordered by the *NetFlows* ranking rule.

```
>>> t.showHTMLPerformanceHeatmap(Correlations=True,rankingRule='NetFlows
→')
```

Heatmap of Performance Tableau 'random3ObjectivesPerfTab'

criteria	ec4	so2	ec1	ec8	en9	en3	so7	en6	en5
weights	+8.00	+12.00	+8.00	+8.00	+6.00	+6.00	+12.00	+6.00	+6.00
tau(*)	+0.55	+0.52	+0.45	+0.38	+0.31	+0.29	+0.24	+0.05	+0.05
p4	72.26	49.05	85.19	44.62	50.92	80.81	55.57	66.21	49.69
p2	86.00	89.12	89.77	89.43	26.65	20.79	84.73	18.66	23.83
p3	68.70	58.85	80.19	91.94	14.85	46.78	15.49	1.65	27.25
p7	36.31	68.06	15.33	38.31	13.10	30.70	58.45	69.71	35.52
p1	22.65	22.43	38.11	77.02	44.37	58.16	28.41	11.41	31.40
p6	75.76	48.59	21.06	29.63	32.96	12.54	26.40	36.04	43.09
p5	8.96	12.36	46.75	35.91	9.83	31.05	44.92	31.22	29.52

Color legend:

quantile	14.29%	28.57%	42.86%	57.14%	71.43%	85.71%	100.00%
----------	--------	--------	--------	--------	--------	--------	---------

(*) tau: Ordinal (Kendall) correlation between marginal criterion and global ranking relation

Outranking model: **standard**, Ranking rule: **NetFlows**

Ordinal (Kendall) correlation between global ranking and global outranking relation: **+0.942**

Mean marginal correlation (a) : **+0.338**

Fig. 5.2: Heat map of the random 3 objectives performance tableau ordered by the *NetFlows* ranking rule

As the initial prekernel is here validated at stability level +2, recommending alternatives *p4*, as well as *p2*, as potential first choices, appears well justified. Alternative *a4* represents indeed an overall *best compromise choice* between all decision objectives, whereas alternative *p2* gives an unanimous best choice with respect to two out of three decision objectives. Up to the decision maker to make his final choice.

For concluding, let us mention that it is precisely again our bipolar-valued *logical characteristic framework* that provides us here with a **first order distributional dominance** test for effectively qualifying the stability level 2 *robustness* of an outranking digraph when facing performance tableaux with criteria of only ordinal-valued significances. A real world application of our stability analysis with such a kind of performance tableau may be consulted in [BIS-2015p].

Back to *Content Table* (page 2)

6 On unopposed outrankings with multiple decision objectives

- *Characterising unopposed multiobjective outranking situations* (page 48)
- *Computing Pareto efficient multiobjective choices* (page 51)

When facing a performance tableau involving multiple decision objectives, the robustness level ± 3 , introduced in the previous Section, may lead to distinguishing what we call **unopposed** outranking situations, like the one shown between alternative p_4 and p_1 ($r(p_4 \succsim p_1) = +0.78$, see Listing 5.4 Line11), namely preferential situations that are more or less validated or invalidated by all the decision objectives.

6.1 Characterising unopposed multiobjective outranking situations

Formally, we say that decision alternative x **outranks** decision alternative y **unopposed** when

- x positively outranks y on one or more decision objective without x being positively outranked by y on any decision objective.

Dually, we say that decision alternative x **does not outrank** decision alternative y **unopposed** when

- x is positively outranked by y on one or more decision objective without x outranking y on any decision objective.

Let us reconsider, for instance, the previous performance tableau with three decision objectives (see Listing 5.1):

Listing 6.1: Performance tableau with three decision objectives

```
1 >>> from randomPerfTabs import\  
2 ...     Random3ObjectivesPerformanceTableau  
3  
4 >>> t = Random3ObjectivesPerformanceTableau(  
5 ...     numberOfActions=7,\br/>6 ...     numberOfCriteria=9,seed=102)  
7  
8 >>> t.showObjectives()  
9 *----- show objectives -----"  
10 Eco: Economical aspect  
11 ec1 criterion of objective Eco 8  
12 ec4 criterion of objective Eco 8  
13 ec8 criterion of objective Eco 8  
14 Total weight: 24.00 (3 criteria)
```

(continues on next page)

(continued from previous page)

```
15 Soc: Societal aspect
16   so2 criterion of objective Soc 12
17   so7 criterion of objective Soc 12
18   Total weight: 24.00 (2 criteria)
19 Env: Environmental aspect
20   en3 criterion of objective Env 6
21   en5 criterion of objective Env 6
22   en6 criterion of objective Env 6
23   en9 criterion of objective Env 6
24   Total weight: 24.00 (4 criteria)
```

We notice in this example three decision objectives of equal importance (see Listing 6.1 Lines 10,15,19). What will be the outranking situations that are positively (resp. negatively) validated for each one of the decision objectives taken individually ?

We may obtain such *unopposed multiobjective* outranking situations by operating an **epistemic o-average fusion** (see the `~digraphsTools.symmetricAverage` method) of the marginal outranking digraphs restricted to the coalition of criteria supporting each one of the decision objectives (see Listing 6.2 below).

Listing 6.2: Computing unopposed outranking situations

```
1 >>> from outrankingDigraphs import BipolarOutrankingDigraph
2 >>> geco = BipolarOutrankingDigraph(t,objectivesSubset=['Eco'])
3 >>> gsoc = BipolarOutrankingDigraph(t,objectivesSubset=['Soc'])
4 >>> genv = BipolarOutrankingDigraph(t,objectivesSubset=['Env'])
5 >>> from digraphs import FusionLDigraph
6 >>> objectiveWeights = \
7 ...   [t.objectives[obj]['weight'] for obj in t.objectives]
8
9 >>> uopg = FusionLDigraph([geco,gsoc,genv],\
10 ...                       operator='o-average',\
11 ...                       weights=objectiveWeights)
12
13 >>> uopg.showRelationTable(ReflexiveTerms=False)
14 * ---- Relation Table ----
15 r   | 'p1'  'p2'  'p3'  'p4'  'p5'  'p6'  'p7'
16 ----|-----
17 'p1' |  -    +0.00 +0.00 -0.69 +0.39 +0.11 +0.00
18 'p2' | +0.00  -    +0.83 +0.00 +0.00 +0.00 +0.00
19 'p3' | +0.00 -0.33  -    +0.00 +0.50 +0.00 +0.00
20 'p4' | +0.78 +0.00 +0.61  -    +1.00 +1.00 +0.67
21 'p5' | -0.11 +0.00 +0.00 -0.89  -    +0.11 +0.00
22 'p6' | +0.00 +0.00 +0.00 -0.44 +0.17  -    +0.00
23 'p7' | +0.00 +0.00 +0.00 +0.00 +0.78 +0.42  -
24 Valuation domain: [-1.000; 1.000]
```

Positive (resp. negative) $r(x \succsim y)$ characteristic values, like $r(p1 \succsim p5) = 0.39$ (see Listing

6.2 Line 17), show hence only outranking situations being validated (resp. invalidated) by one or more decision objectives without being invalidated (resp. validated) by any other decision objective.

For easily computing this kind of *unopposed multiobjective* outranking digraphs, the `outrankingDigraphs` module conveniently provides a corresponding `UnOpposedBipolarOutrankingDigraph` constructor.

Listing 6.3: Unopposed outranking digraph constructor

```

1  >>> from outrankingDigraphs import\
2  ...     UnOpposedBipolarOutrankingDigraph
3
4  >>> uopg = UnOpposedBipolarOutrankingDigraph(t)
5  >>> uopg
6  *----- Object instance description -----*
7  Instance class      : UnOpposedBipolarOutrankingDigraph
8  Instance name      : unopposed_outrankings
9  # Actions          : 7
10 # Criteria          : 9
11 Size               : 13
12 Oppositeness (%)    : 43.48
13 Determinateness (%) : 61.71
14 Valuation domain   : [-1.00;1.00]
15 Attributes         : ['name', 'actions', 'valuationdomain', 'objectives
16                      ↪',
17                      'criteria', 'methodData', 'evaluation', 'order',
18                      'runTimes', 'relation',
19                      ↪ 'marginalRelationsRelations',
20                      'gamma', 'notGamma']
21 >>> uopg.computeOppositeness(InPercents=True)
22 {'standardSize': 23, 'unopposedSize': 13,
23  'oppositeness': 43.47826086956522}

```

The resulting *unopposed* outranking digraph keeps in fact 13 (see Listing 6.3 Lines 12-13) out of the 23 positively validated *standard* outranking situations, leading to a degree of **oppositeness**-preferential disagreement between decision objectives- of $(1.0 - 13/23) = 0.4348$.

We may now, for instance, verify the unopposed status of the outranking situation observed between alternatives *p1* and *p5*.

Listing 6.4: Example of unopposed multiobjective outranking situation

```

1  >>> uopg.showPairwiseComparison('p1', 'p5')
2  *----- pairwise comparison -----*
3  Comparing actions : (p1, p5)
4  crit. wght.  g(x)  g(y)  diff  | ind  pref  r()  |
5  ec1    8.00  38.11  46.75  -8.64  | 5.00  10.00  +0.00  |

```

(continues on next page)

(continued from previous page)

6	ec4	8.00	22.65	8.96	+13.69		5.00	10.00	+8.00	
7	ec8	8.00	77.02	35.91	+41.11		5.00	10.00	+8.00	
8	en3	6.00	58.16	31.05	+27.11		5.00	10.00	+6.00	
9	en5	6.00	31.40	29.52	+1.88		5.00	10.00	+6.00	
10	en6	6.00	11.41	31.22	-19.81		5.00	10.00	-6.00	
11	en9	6.00	44.37	9.83	+34.54		5.00	10.00	+6.00	
12	so2	12.00	22.43	12.36	+10.07		5.00	10.00	+12.00	
13	so7	12.00	28.41	44.92	-16.51		5.00	10.00	-12.00	
14	Valuation in range: -72.00 to +72.00; global concordance: +28.00									

In Listing 6.4 we see that alternative $p1$ does indeed positively outrank alternative $p5$ from the economic perspective ($r(p1 \succ_{Eco} p5) = +16/24$) as well as from the environmental perspective ($r(p1 \succ_{Env} p5) = +12/24$). Whereas, from the societal perspective, both alternatives appear incomparable ($r(p1 \succ_{Soc} p5) = 0/24$).

When fixed proportional criteria significances per objective are given, these outranking situations appear hence **stable** with respect to all possible importance weights we could allocate to the decision objectives.

This gives way for computing multiobjective **Pareto efficient** choice recommendations.

6.2 Computing *Pareto* efficient multiobjective choices

Indeed, best choice recommendations, computed from an *unopposed multiobjective* outranking digraph, will in fact deliver **Pareto efficient** choices.

Listing 6.5: Pareto efficient multiobjective choice recommendation

```
1 >>> uopg.showBestChoiceRecommendation()
2 Best choice recommendation(s) (BCR)
3 (in decreasing order of determinateness)
4 Credibility domain: [-1.00,1.00]
5 == >> potential first choice(s)
6 choice : ['p2', 'p4', 'p7']
7 independence : 0.00
8 dominance : 0.33
9 absorbency : 0.00
10 covering (%) : 33.33
11 determinateness (%) : 50.00
12 == >> potential last choice(s)
13 choice : ['p3', 'p5', 'p6', 'p7']
14 independence : 0.00
15 dominance : -0.61
16 absorbency : 0.11
17 covered (%) : 33.33
18 determinateness (%) : 50.00
```

Our previous *robust best choice recommendation* ($p2$ and $p4$, see Fig. 5.1) remains, in this example here, **stable**. We recover indeed the best choice recommendation [$p2$, $p4$, $p7$] (see Listing 6.5 Line 6). Yet, notice that decision alternative $p7$ appears to be at the same time a potential *first* as well as a potential *last* choice recommendation (see Line 13), a consequence of $p7$ being completely *incomparable* to the other decision alternatives when restricting the comparability to only unopposed strict outranking situations.

We may visualize this kind of **Pareto efficient** result in Fig. 6.1 below.

```
>>> (~(-uopg)).exportGraphViz(fileName = 'unopDigraph',\
...                               firstChoice = ['p2', 'p4'],\
...                               lastChoice = ['p3', 'p5', 'p6'])
```

—- *exporting a dot file for GraphViz tools* —- Ex-
 porting to unopDigraph.dot dot -Grankdir=BT -Tpng
 unopDigraph.dot -o unopDigraph.png

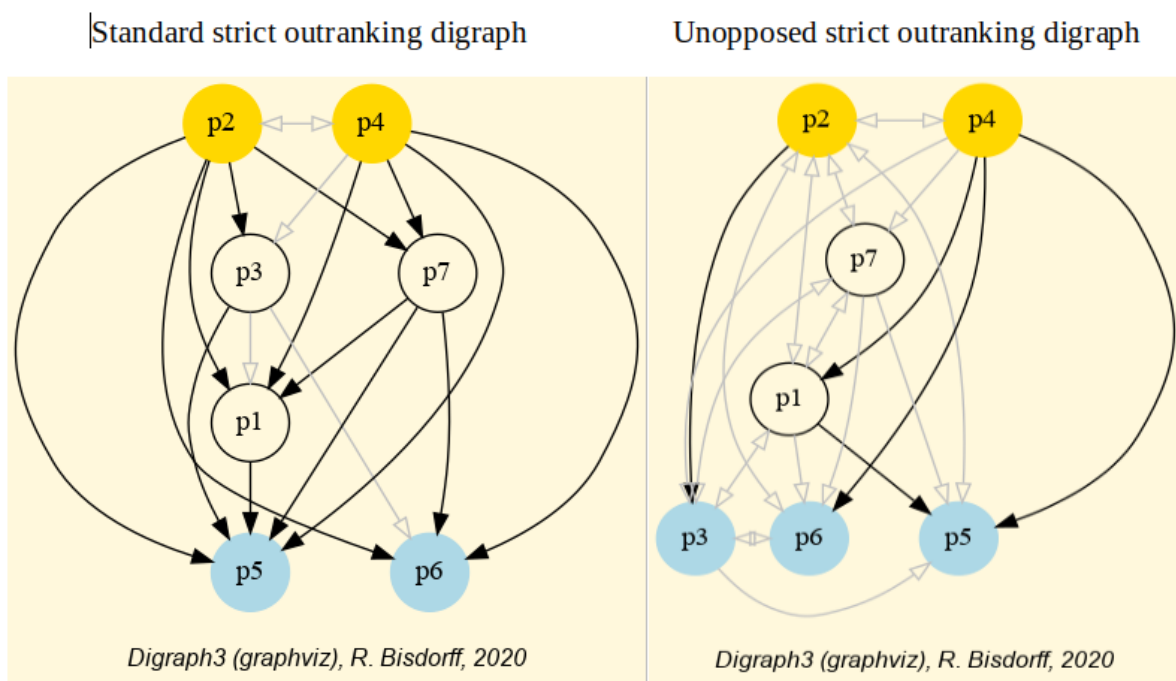


Fig. 6.1: Standard versus *unopposed* strict outranking digraphs oriented by first and last choice recommendations

In order to make now an eventual best unique choice, a decision maker will necessarily have to weight, in a second stage of the decision aiding process, the relative importance of the individual decision objectives (see tutorial on computing a best choice recommendation).

Back to [Content Table](#) (page 2)

7 Two-stage elections with multipartisan primary selection

- *Converting voting profiles into performance tableaux* (page 53)
- *Multipartisan primary selection of eligible candidates* (page 55)
- *Secondary election winner determination* (page 57)
- *Multipartisan preferences in divisive politics* (page 58)

Let us finally remark that in a *social choice* context, where decision objectives would match different political parties, *Pareto efficient choice recommendations* represent in fact **multipartisan social choices** that could judiciously deliver the primary selection in a two stage election system.

To compute such *Pareto* efficient social choices we need to, first, convert a given linear voting profile (with polls) into a corresponding performance tableau.

7.1 Converting voting profiles into performance tableaux

We shall illustrate this point with a voting profile we discuss in the tutorial on generating random linear voting profiles.

Listing 7.1: Example of a 3 parties voting profile

```
1  >>> from votingProfiles import RandomLinearVotingProfile
2  >>> lvp = RandomLinearVotingProfile(numberOfCandidates=15,
3  ...                                numberOfVoters=1000,
4  ...                                WithPolls=True,
5  ...                                partyRepartition=0.5,
6  ...                                other=0.1,
7  ...                                seed=0.9189670954954139)
8
9  >>> lvp
10 *----- VotingProfile instance description -----*
11 Instance class      : RandomLinearVotingProfile
12 Instance name       : randLinearProfile
13 # Candidates        : 15
14 # Voters            : 1000
15 Attributes          : ['name', 'seed', 'candidates',
16                        'voters', 'WithPolls', 'RandomWeights',
17                        'sumWeights', 'poll1', 'poll2',
18                        'other', partyRepartition,
19                        'linearBallot', 'ballot']
20 >>> lvp.showRandomPolls()
21 Random repartition of voters
```

(continues on next page)

(continued from previous page)

```
22 Party_1 supporters : 460 (46.0%)
23 Party_2 supporters : 436 (43.6%)
24 Other voters      : 104 (10.4%)
25 *----- random polls -----*
26 Party_1(46.0%) | Party_2(43.6%) | expected
27 -----
28 a06 : 19.91% | a11 : 22.94% | a06 : 15.00%
29 a07 : 14.27% | a08 : 15.65% | a11 : 13.08%
30 a03 : 10.02% | a04 : 15.07% | a08 : 09.01%
31 a13 : 08.39% | a06 : 13.40% | a07 : 08.79%
32 a15 : 08.39% | a03 : 06.49% | a03 : 07.44%
33 a11 : 06.70% | a09 : 05.63% | a04 : 07.11%
34 a01 : 06.17% | a07 : 05.10% | a01 : 05.06%
35 a12 : 04.81% | a01 : 05.09% | a13 : 05.04%
36 a08 : 04.75% | a12 : 03.43% | a15 : 04.23%
37 a10 : 04.66% | a13 : 02.71% | a12 : 03.71%
38 a14 : 04.42% | a14 : 02.70% | a14 : 03.21%
39 a05 : 04.01% | a15 : 00.86% | a09 : 03.10%
40 a09 : 01.40% | a10 : 00.44% | a10 : 02.34%
41 a04 : 01.18% | a05 : 00.29% | a05 : 01.97%
42 a02 : 00.90% | a02 : 00.21% | a02 : 00.51%
```

In this example (see `linearVotingProfileWithPolls` Lines 18-), we obtained 460 *Party_1* supporters (46%), 436 *Party_2* supporters (43.6%) and 104 other voters (10.4%). Favorite candidates of *Party_1* supporters, with more than 10%, appeared to be *a06* (19.91%), *a07* (14.27%) and *a03* (10.02%). Whereas for *Party_2* supporters, favorite candidates appeared to be *a11* (22.94%), followed by *a08* (15.65%), *a04* (15.07%) and *a06* (13.4%).

We may convert this linear voting profile into a `PerformanceTableau` object where each party corresponds to a decision objective.

Listing 7.2: Converting a voting profile into a performance tableau

```
1 >>> lvp.save2PerfTab('votingPerfTab')
2 >>> from perfTabs import PerformanceTableau
3 >>> vpt = PerformanceTableau('votingPerfTab')
4 >>> vpt
5 *----- PerformanceTableau instance description -----*
6 Instance class      : PerformanceTableau
7 Instance name       : votingPerfTab
8 # Actions           : 15
9 # Objectives         : 3
10 # Criteria          : 1000
11 Attributes          : ['name', 'actions', 'objectives',
12                        'criteria', 'weightPreorder', 'evaluation']
```

(continues on next page)

(continued from previous page)

```
13 >>> vpt.objectives
14 OrderedDict([
15   ('party0', {'name': 'other', 'weight': Decimal('104'),
16   'criteria': ['v0003', 'v0008', 'v0011', ... ]}),
17   ('party1', {'name': 'party 1', 'weight': Decimal('460'),
18   'criteria': ['v0002', 'v0006', 'v0007', ... ]}),
19   ('party2', {'name': 'party 2', 'weight': Decimal('436'),
20   'criteria': ['v0001', 'v0004', 'v0005', ... ]})
21 ])
```

In Listing 7.2 we first store the linear voting in a `PerformanceTableau` format (see Line 1). In Line 3, we reload this performance tableau data. The three parties of the linear voting profile represent three decision objectives and the voters are distributed as performance criteria according to the party they support.

7.2 Multipartisan primary selection of eligible candidates

In order to make now a **primary multipartisan selection** of potential election winners, we compute the corresponding *unopposed multiobjective outranking* digraph.

Listing 7.3: Computing unopposed multiobjective outranking situations

```
1 >>> from outrankingDigraphs import \
2 ...     UnOpposedBipolarOutrankingDigraph
3
4 >>> uog = UnOpposedBipolarOutrankingDigraph(vpt)
5 >>> uog
6 *----- Object instance description -----*
7 Instance class      : UnOpposedBipolarOutrankingDigraph
8 Instance name      : unopposed_outrankings
9 # Actions          : 15
10 # Criteria         : 1000
11 Size               : 34
12 Oppositeness (%)   : 67.31
13 Determinateness (%) : 57.61
14 Valuation domain   : [-1.00;1.00]
15 Attributes         : ['name', 'actions', 'valuationdomain',
16                       'objectives', 'criteria', 'methodData',
17                       'evaluation', 'order', 'runTimes', '
18                       relation', 'marginalRelationsRelations',
19                       'gamma', 'notGamma']
```

From the potential 105 pairwise outranking situations, we keep 34 positively validated outranking situations, leading to a degree of *oppositeness* between political parties of 67.31%.

We may visualize the corresponding bipolar-valued relation table by orienting the list of candidates with the help of the *initial* and the *terminal prekernels*.

Listing 7.4: Visualizing the unopposed outranking relation

```

1  >>> uog.showPreKernels()
2  *--- Computing preKernels ---*
3  Dominant preKernels :
4  ['a11', 'a06', 'a13', 'a15']
5      independence : 0.0
6      dominance    : 0.18
7      absorbency   : -0.66
8      covering     : 0.43
9  Absorbent preKernels :
10 ['a02', 'a04', 'a14', 'a03']
11     independence : 0.0
12     dominance    : 0.0
13     absorbency   : 0.37
14     covered      : 0.46
15 >>> orientedCandidatesList = ['a06','a11','a13','a15',\
16 ...                          'a01','a05','a07','a08','a09','a10','a12',\
17 ...                          'a02','a03','a04','a14']
18
19 >>> uog.showHTMLRelationTable(\
20 ...     actionsList=orientedCandidatesList,\
21 ...     tableTitle='Unopposed three-partisan outrankings')
```

Multipartisan outranking situations

r(x S y)	a06	a11	a13	a15	a01	a05	a07	a08	a09	a10	a12	a02	a03	a04	a14
a06	-	0.00	0.00	0.00	0.44	0.00	0.25	0.00	0.00	0.00	0.56	0.86	0.29	0.00	0.57
a11	0.00	-	0.00	0.00	0.00	0.55	0.00	0.18	0.59	0.51	0.39	0.80	0.00	0.42	0.47
a13	0.00	0.00	-	0.00	0.00	0.52	-0.27	0.00	0.00	0.00	0.00	0.77	0.00	0.00	0.16
a15	0.00	0.00	0.00	-	0.00	0.39	0.00	0.00	0.00	0.00	0.00	0.66	0.00	0.00	0.00
a01	-0.44	0.00	0.00	0.00	-	0.00	0.00	0.00	0.00	0.00	0.00	0.77	0.00	0.00	0.20
a05	0.00	-0.55	-0.52	-0.39	0.00	-	0.00	-0.47	0.00	-0.12	0.00	0.37	0.00	0.00	0.00
a07	-0.25	0.00	0.27	0.00	0.00	0.00	-	0.00	0.00	0.00	0.30	0.83	0.00	0.00	0.38
a08	0.00	-0.18	0.00	0.00	0.00	0.47	0.00	-	0.00	0.00	0.00	0.77	0.00	0.29	0.00
a09	0.00	-0.59	0.00	0.00	0.00	0.00	0.00	0.00	-	0.00	0.00	0.55	0.00	0.00	0.00
a10	0.00	-0.51	0.00	0.00	0.00	0.12	0.00	0.00	0.00	-	0.00	0.50	0.00	0.00	0.00
a12	-0.56	-0.39	0.00	0.00	0.00	0.00	-0.30	0.00	0.00	0.00	-	0.72	0.00	0.00	0.10
a02	-0.86	-0.80	-0.77	-0.66	-0.77	-0.37	-0.83	-0.77	-0.55	-0.50	-0.72	-	0.00	0.00	0.00
a03	-0.29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-	0.00	0.00
a04	0.00	-0.42	0.00	0.00	0.00	0.00	0.00	-0.29	0.00	0.00	0.00	0.00	0.00	-	0.00
a14	-0.57	-0.47	-0.16	0.00	-0.20	0.00	-0.38	0.00	0.00	0.00	-0.10	0.00	0.00	0.00	-

Valuation domain: [-1.00; +1.00]

Fig. 7.1: Relation table of multipartisan outranking digraph

In Fig. 7.1, we may notice that the dominating outranking prekernel ['a06', 'a11', 'a13', 'a15'] gathers in fact a **multipartisan selection** of potential election winners. It is worthwhile noticing that in Fig. 7.1 the majority margins obtained from a linear voting profile do verify the zero-sum rule ($r(x \succsim y) + r(y \succsim x) = 0.0$). To each positive outranking situation corresponds indeed an equivalent negative converse situation and the resulting outranking and strict outranking digraphs are the same.

7.3 Secondary election winner determination

When restricting now, in a secondary election stage, the set of eligible candidates to this dominating prekernel, we may compute the actual best social choice.

Listing 7.5: Secondary election winner recommendation

```

1 >>> from outrankingDigraphs import BipolarOutrankingDigraph
2 >>> g2 = BipolarOutrankingDigraph(vpt,\
3 ...     actionsSubset=['a06','a11','a13','a15'])
4
5 >>> g2.showRelationTable(ReflexiveTerms=False)
6 * ---- Relation Table ----
7   r      | 'a06'  'a11'  'a13'  'a15'
8   .-----|-----
9   'a06' |    -    +0.10 +0.48 +0.52

```

(continues on next page)

(continued from previous page)

```
10  'a11' | -0.10    -    +0.27  +0.29
11  'a13' | -0.48  -0.27    -    +0.19
12  'a15' | -0.52  -0.29  -0.19    -
13  Valuation domain: [-1.000; 1.000]
14  >>> g2.computeCondorcetWinners()
15  ['a06']
16  >>> g2.computeCopelandRanking()
17  ['a06', 'a11', 'a13', 'a15']
```

Candidate *a06* appears clearly to be the winner of this election. Notice by the way that the restricted pairwise outranking relation shown in [Listing 7.5](#) represents a linear ordering of the preselected candidates.

We may eventually check the quality of this best choice by noticing that candidate *a06* represents indeed the *simple majority* winner, the *instant-run-off* winner, the *Borda*, as well as the *Condorcet winner* of the initially given linear voting profile *lvp* (see [Listing 7.1](#)).

Listing 7.6: Secondary election winner recommendation verification

```
1  >>> lvp.computeSimpleMajorityWinner()
2  ['a06']
3  >>> lvp.computeInstantRunoffWinner()
4  ['a06']
5  >>> lvp.computeBordaWinners()
6  ['a06']
7  >>> from votingProfiles import MajorityMarginsDigraph
8  >>> cd = MajorityMarginsDigraph(lvp)
9  >>> cd.computeCondorcetWinners()
10 ['a06']
```

In our example voting profile here, the multipartisan primary selection stage appears quite effective in reducing the number of eligible candidates to four out of a set of 15 candidates without btw rejecting the actual winning candidate.

7.4 Multipartisan preferences in divisive politics

However, in a very **divisive two major party system**, like in the US, where preferences of the supporters of one party appear to be very opposite to the preferences of the supporters of the other major party, the multipartisan outranking digraph will become nearly indeterminate.

In [Listing 7.7](#) below we generate such a divisive kind of linear voting profile with the help of the *DivisivePolitics* flag⁵ (see Lines 4 and 13-19). When now converting the voting

⁵ The `RandomLinearVotingProfile` constructor provides a *DivisivePolitics* flag (*False* by default) for generating random linear voting profiles based on a divisive polls structure.

profile into a performance tableau (Lines 20-21), we may compute the corresponding unopposed outranking digraph.

Listing 7.7: A divisive two-party example of a random linear voting profile

```

1  >>> from votingProfiles import RandomLinearVotingProfile
2  >>> lvp = RandomLinearVotingProfile(\
3  ...     numberOfCandidates=7,numberOfVoters=500,\
4  ...     WithPolls=True, partyRepartition=0.4,other=0.2,\
5  ...     DivisivePolitics=True, seed=1)
6
7  >>> lvp.showRandomPolls()
8  Random repartition of voters
9  Party_1 supporters : 240 (48.00%)
10 Party_2 supporters : 160 (32.00%)
11 Other voters       : 100 (20.00%)
12 *----- random polls -----*
13 Party_1(48.0%) | Party_2(32.0%) | expected
14 -----
15 a2 : 30.84% | a1 : 30.84% | a2 : 15.56%
16 a3 : 23.67% | a4 : 23.67% | a3 : 12.91%
17 a7 : 17.29% | a6 : 17.29% | a7 : 11.43%
18 a5 : 11.22% | a5 : 11.22% | a1 : 11.00%
19 a6 : 09.79% | a7 : 09.79% | a6 : 10.23%
20 a4 : 04.83% | a3 : 04.83% | a4 : 09.89%
21 a1 : 02.37% | a2 : 02.37% | a5 : 08.98%
22 >>> lvp.save2PerfTab('divisiveExample')
23 >>> dvp = PerformanceTableau('divisiveExample')
24 >>> from outrankingDigraphs import \
25 ...     UnOpposedBipolarOutrankingDigraph
26
27 >>> uodg = UnOpposedBipolarOutrankingDigraph(dvp)
28 >>> uodg
29 *----- Object instance description -----*
30 Instance class      : UnOpposedBipolarOutrankingDigraph
31 Instance name       : unopposed_outrankings
32 # Actions           : 7
33 # Criteria           : 500
34 Size                : 0
35 Oppositeness (%)    : 100.00
36 Determinateness (%) : 50.00
37 Valuation domain    : [-1.00;1.00]

```

With an oppositeness degree of 100.0% (see Listing 7.7 Lines 33-34), the preferential disagreement between the political parties is complete, and the unopposed outranking digraph *uodg* becomes completely **indeterminate** as shown in the relation table below.

```

1 >>> uodg.showRelationTable(ReflexiveTerms=False)
2 * ---- Relation Table ----
3   r   |   'a1'   'a2'   'a3'   'a4'   'a5'   'a6'   'a7'
4   ----|-----
5 'a1' |      -    +0.00  +0.00  +0.00  +0.00  +0.00  +0.00
6 'a2' |   +0.00     -    +0.00  +0.00  +0.00  +0.00  +0.00
7 'a3' |   +0.00  +0.00     -    +0.00  +0.00  +0.00  +0.00
8 'a4' |   +0.00  +0.00  +0.00     -    +0.00  +0.00  +0.00
9 'a5' |   +0.00  +0.00  +0.00  +0.00     -    +0.00  +0.00
10 'a6' |   +0.00  +0.00  +0.00  +0.00  +0.00     -    +0.00
11 'a7' |   +0.00  +0.00  +0.00  +0.00  +0.00  +0.00     -
12 Valuation domain: [-1.000; 1.000]

```

As a consequence, a **multipartisan primary selection**, computed with a `showBestChoiceRecommendation()` method, will keep the complete initial set of eligible candidates and, hence, becomes **ineffective** (see [Listing 7.8](#) Line 6).

Listing 7.8: Example of ineffective primary multipartisan selection

```

1 >>> uodg.showBestChoiceRecommendation()
2 Rubis best choice recommendation(s) (BCR)
3   (in decreasing order of determinateness)
4 Credibility domain: [-1.00,1.00]
5 == >> ambiguous choice(s)
6 choice           : ['a1','a2','a3','a4','a5','a6','a7']
7 independence      : 0.00
8 dominance         : 1.00
9 absorbency        : 1.00
10 covered (%)       : 100.00
11 determinateness (%) : 50.00
12 - most credible action(s) = { }

```

With such kind of divisive voting profile, there may not always exist an obvious winner. In [Listing 7.9](#) below, we see, for instance, that the *simple majority* winner is *a2* (Line 2), whereas the *instant-run-off* winner is *a6* (Line 4).

Listing 7.9: Example of secondary selection

```

1 >>> lvp.computeSimpleMajorityWinner()
2 ['a2']
3 >>> lvp.computeInstantRunoffWinner()
4 ['a6']
5 >>> from votingProfiles import MajorityMarginsDigraph
6 >>> cg = MajorityMarginsDigraph(lvp)
7 >>> cg.showRelationTable(ReflexiveTerms=False)
8 * ---- Relation Table ----
9   S   |   'a1' 'a2' 'a3' 'a4' 'a5' 'a6' 'a7'

```

(continues on next page)

(continued from previous page)

```
10  -----|-----
11  'a1' |  -   -68  -90  -46  -68  -88  -84
12  'a2' | +68   -   -32  +80  +46   -  -24
13  'a3' | +90  +32   -   +58  +46   +4   +8
14  'a4' |  +4  -80  -58   -  -16  -68  -72
15  'a5' | +68  -46  -46  +16   -  -26  -64
16  'a6' | +88   +6   -4  +68  "26   -   -2
17  'a7' | +84  +24   -8  +72  "64   "2   -
18  Valuation domain: [-500;+500]
19  >>> cg.computeCondorcetWinners()
20  ['a3']
21  >>> lvp.computeBordaWinners()
22  ['a3', 'a7']
23  >>> cg.computeCopelandRanking()
24  ['a3', 'a7', 'a6', 'a2', 'a5', 'a4', 'a1']
```

But in our example here, we are lucky. When constructing with the *pairwise majority margins* digraph (Line 6), a *Condorcet winner*, namely *a3* becomes apparent (Lines 13,20), which is also one of the two *Borda* winners (Line 22). More interesting even is to notice that the apparent majority margins digraph models in fact a linear ranking $[a3, a7, a6, a2, a5, a4, a1]$ of all the eligible candidates, as shown with a Copeland ranking rule (Line 24).

We may eventually visualize in [Listing 7.10](#) this linear ranking with a graphviz drawing where we drop all transitive arcs (Line 1) and orient the drawing with *Condorcet* winner *a3* and loser *a1* (Lines 2).

Listing 7.10: Drawing the linear ordering

```
1 >>> cg.closeTransitive(Reverse=True)
2 >>> cg.exportGraphViz('divGraph',firstChoice=['a3'],lastChoice=['a1'])
3 *---- exporting a dot file for GraphViz tools -----*
4 Exporting to divGraph.dot
5 dot -Grankdir=BT -Tpng divGraph.dot -o divGraph.png
```

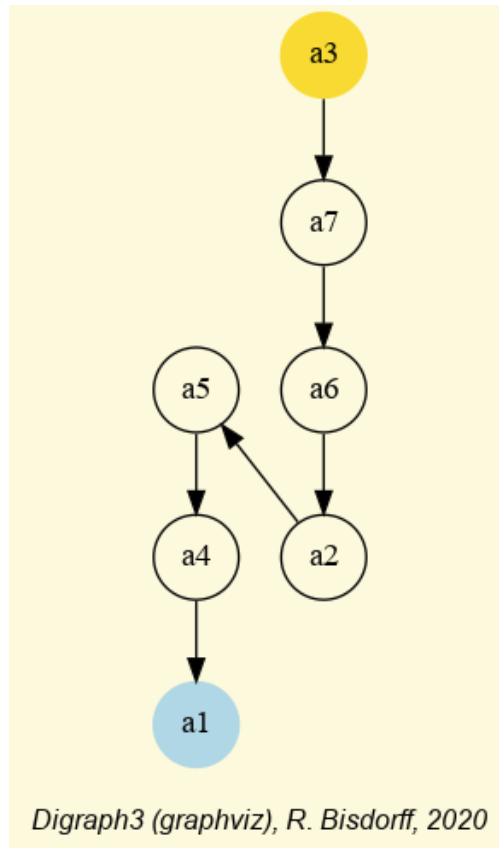


Fig. 7.2: Linear ordering of the eligible candidates

Back to *Content Table* (page 2)

8 Tempering plurality tyranny effects with bipolar approval voting

The choice of a voting procedure shapes the democracy in which we live.

—Baujard A., Gavrel F., Igersheim H., Laslier J.-F. and Lebon I.
[BAU-2013p].

- *Bipolar approval voting systems* (page 63)
- *Pairwise comparison of bipolar approval votes* (page 66)
- *Three-valued evaluative voting system* (page 68)
- *Favouring multipartisan candidates* (page 71)

8.1 Bipolar approval voting systems

In the `votingProfiles` module we provide a `BipolarApprovalVotingProfile` class for handling voting results where, for each eligible candidate c , the voters are invited to **approve** (+1), **disapprove** (-1), or **ignore** (0) the statement that *candidate C should win the election*.

File `bpApVotingProfile.py` contains such a bipolar approval voting profile concerning 100 voters and 15 eligible candidates. We may inspect its content as follows.

```

1 >>> from votingProfiles import *
2 >>> bavp = BipolarApprovalVotingProfile('bpApVotingProfile')
3 >>> bavp
4 *----- VotingProfile instance description -----*
5 Instance class      : BipolarApprovalVotingProfile
6 Instance name       : bpApVotingProfile
7 # Candidates        : 15
8 # Voters             : 100
9 Attributes          : ['name', 'candidates', 'voters',
10                        'approvalBallot', 'netApprovalScores',
11                        'ballot']

```

Beside the `bavp.candidates` and `bavp.voters` attributes, we discover in Line 10 above the `bavp.approvalBallot` attribute which gathers bipolar approval votes. Its content is the following.

Listing 8.1: Inspecting a bipolar approval ballot

```

1 >>> bavp.approvalBallot
2 {'v001':
3   {'a01': Decimal('0'),
4     ...
5     'a04': Decimal('1'),
6     ...
7     'a15': Decimal('0')}
8 },
9 'v002':
10 {'a01': Decimal('-1'),
11   'a02': Decimal('0'),
12   ...

```

(continues on next page)

(continued from previous page)

```
13     'a15': Decimal('1')
14 },
15 ...
16 v100':
17 {'a01': Decimal('0'),
18  'a02': Decimal('1'),
19  ...
20  'a15': Decimal('1')
21 }
22 }
```

Let us denote A_v the set of candidates approved by voter v . In Listing 8.1 we hence record in fact the bipolar-valued truth characteristic values $r(c \in A_v)$ of the statements that candidate c is **approved** by voter v . In Line 5, we observe for instance that voter $v001$ **positively approves** candidate $a04$. And, in Line 10, we see that voter $v002$ **negatively approves**, i.e. **positively disapproves** candidate $a01$. We may now consult how many approvals or disapprovals each candidate receives.

```
1 >>> bavp.showApprovalResults()
2 Approval results
3 Candidate: a12 obtains 34 votes
4 Candidate: a05 obtains 30 votes
5 Candidate: a03 obtains 28 votes
6 Candidate: a14 obtains 27 votes
7 Candidate: a11 obtains 27 votes
8 Candidate: a04 obtains 27 votes
9 Candidate: a01 obtains 27 votes
10 Candidate: a13 obtains 24 votes
11 Candidate: a07 obtains 24 votes
12 Candidate: a15 obtains 23 votes
13 Candidate: a02 obtains 23 votes
14 Candidate: a09 obtains 22 votes
15 Candidate: a08 obtains 22 votes
16 Candidate: a10 obtains 21 votes
17 Candidate: a06 obtains 21 votes
18 Total approval votes: 380
19 Approval proportion: 380/1500 = 0.25
20 >>> bavp.showDisapprovalResults()
21 Disapproval results
22 Candidate: a12 obtains 16 votes
23 Candidate: a03 obtains 22 votes
24 Candidate: a09 obtains 23 votes
25 Candidate: a04 obtains 24 votes
26 Candidate: a06 obtains 24 votes
27 Candidate: a13 obtains 24 votes
28 Candidate: a11 obtains 25 votes
```

(continues on next page)

(continued from previous page)

```
29 Candidate: a02 obtains 26 votes
30 Candidate: a07 obtains 26 votes
31 Candidate: a08 obtains 26 votes
32 Candidate: a05 obtains 27 votes
33 Candidate: a10 obtains 27 votes
34 Candidate: a14 obtains 27 votes
35 Candidate: a15 obtains 27 votes
36 Candidate: a01 obtains 32 votes
37 Total disapproval votes: 376
38 Disapproval proportion: 376/1500 = 0.25
```

In Lines 3 and 22 above, we may see that, of all potential candidates, it is Candidate *a12* who receives the highest number of approval votes (34) and the lowest number of disapproval votes (16). Total number of approval, respectively disapproval, votes approaches more or less a proportion of 25% of the $100 \times 15 = 1500$ potential approval votes. About 50% of the latter remain hence ignored.

When operating now, for each candidate *c*, the difference between the number of approval and the number of disapproval votes he receives, we obtain per candidate a corresponding **net approval** score; in fact, the bipolar truth characteristic value of the statement *candidate c should win the election*.

$$r(\text{Candidate } c \text{ should win the election}) = \sum_v (r(c \in A_v))$$

These bipolar characteristic values are stored in the *bavp.netApprovalScores* attribute and may be printed out as follows.

```
1 >>> bavp.showNetApprovalScores()
2 Net Approval Scores
3 Candidate: a12 obtains 18 net approvals
4 Candidate: a03 obtains 6 net approvals
5 Candidate: a05 obtains 3 net approvals
6 Candidate: a04 obtains 3 net approvals
7 Candidate: a11 obtains 2 net approvals
8 Candidate: a14 obtains 0 net approvals
9 Candidate: a13 obtains 0 net approvals
10 Candidate: a09 obtains -1 net approvals
11 Candidate: a07 obtains -2 net approvals
12 Candidate: a06 obtains -3 net approvals
13 Candidate: a02 obtains -3 net approvals
14 Candidate: a15 obtains -4 net approvals
15 Candidate: a08 obtains -4 net approvals
16 Candidate: a01 obtains -5 net approvals
17 Candidate: a10 obtains -6 net approvals
```

We observe in Line 3 above that Candidate *a12*, with a net approval score of $34 - 16 = 18$, represents indeed the **best approved** candidate for winning the election. With a net approval score of $28 - 22 = 6$, Candidate *a03* appears **2nd-best approved**. The net approval scores define hence a potentially weak ranking on the set of eligible elec-

tion candidates, and the winner(s) of the election is(are) determined by the first-ranked candidate(s).

8.2 Pairwise comparison of bipolar approval votes

The approval votes of each voter define now on the set of eligible candidates three ordered categories: his approved (+1), his ignored (0) and his disapproved (-1) ones. Within each of these three categories we consider the voter's actual preferences as **not communicated**, i.e. as *missing data*. This gives for each voter a *partially determined strict order* which we find in the *bavp.ballot* attribute.

```

1 >>> bavp.ballot['v001']['a12']
2 {'a02': Decimal('1'), 'a11': Decimal('1'),
3  'a14': Decimal('1'), 'a04': Decimal('0'),
4  'a06': Decimal('1'), 'a05': Decimal('1'),
5  'a12': Decimal('0'), 'a13': Decimal('0'),
6  'a15': Decimal('1'), 'a01': Decimal('1'),
7  'a08': Decimal('1'), 'a07': Decimal('1'),
8  'a09': Decimal('0'), 'a03': Decimal('1'),
9  'a10': Decimal('0')}
```

For voter *v001*, for instance, the best approved candidate *a12* is strictly preferred to candidates: *a01*, *a02*, *a03*, *a05*, *a06*, *a07*, *a08*, *a11*, *a14* and *15*. No candidate is preferred to *a12* and the comparison with *a04*, *a09*, *a10* and *a13* is not communicated, hence indeterminate. Mind by the way that the reflexive comparison of *a12* with itself is, as usual, is ignored, i.e. indeterminate. Each voter *v* defines thus a partially determined transitive strict preference relation denoted \succ_v on the eligible candidates.

For each pair of eligible candidates, we aggregate the previous individual voter's preferences into a truth characteristic of the statement: candidate *x* is *better approved than* candidate *y*, denoted $r(x \succ y)$

$$r(x \succ y) = \sum_v (r(x \succ_v y)).$$

We say that candidate *x* is *better approved than* Candidate *y* when $r(x \succ y) > 0$, i.e. there is a *majority* of voters who *approve more* and *disapprove less* *x* than *y*. Vice-versa, we say that candidate *x* is *not better approved than* candidate *y* when $r(x \succ y) < 0$, i.e. there is a majority of voters who disapprove more and approve less *x* than *y*. This computation is achieved with the *MajorityMarginsDigraph* constructor.

```

1 >>> from votingProfiles import MajorityMarginsDigraph
2 >>> m = MajorityMarginsDigraph(bavp)
3 >>> m
4 *----- Digraph instance description -----*
5 Instance class      : MajorityMarginsDigraph
6 Instance name       : rel_bpApVotingProfile
7 Digraph Order       : 15
8 Digraph Size        : 97
```

(continues on next page)

(continued from previous page)

```
9 Valuation domain : [-100.00;100.00]
10 Determinateness (%) : 52.55
11 Attributes      : ['name', 'actions', 'criteria',
12                   'ballot', 'valuationdomain', '
13                   relation', 'order',
14                   'gamma', 'notGamma']
```

The resulting digraph m contains 97 positively validated relations (see Line 8 above) and (see Line 9) for all pairs (x, y) of eligible candidates, $r(x \succ y)$ takes value in an valuation range from -100.00 (all voters opposed) to +100.00 (unanimously supported).

We may inspect these pairwise $r(x \succ y)$ values in a browser view.

```
>>> m.showHTMLRelationTable(relationName='r(x > y)')
```

r(x >= y)	a01	a02	a03	a04	a05	a06	a07	a08	a09	a10	a11	a12	a13	a14	a15
a01	-	1	-10	-10	-8	-2	-2	-3	-3	2	-5	-19	-4	-2	2
a02	-1	-	-5	-4	-9	-2	5	1	-7	3	-2	-20	-1	-4	-2
a03	10	5	-	1	4	6	5	10	7	11	0	-5	2	6	9
a04	10	4	-1	-	2	5	7	8	-1	4	0	-7	8	2	8
a05	8	9	-4	-2	-	4	6	7	4	2	1	-17	6	0	3
a06	2	2	-6	-5	-4	-	-1	2	-3	5	-4	-13	-1	-1	2
a07	2	-5	-5	-7	-6	1	-	7	-4	2	-5	-17	-2	-2	4
a08	3	-1	-10	-8	-7	-2	-7	-	-2	2	-2	-16	0	0	-1
a09	3	7	-7	1	-4	3	4	2	-	3	0	-18	-4	0	2
a10	-2	-3	-11	-4	-2	-5	-2	-2	-3	-	-6	-15	-4	-4	2
a11	5	2	0	0	-1	4	5	2	0	6	-	-15	4	0	5
a12	19	20	5	7	17	13	17	16	18	15	15	-	12	13	18
a13	4	1	-2	-8	-6	1	2	0	4	4	-4	-12	-	1	4
a14	2	4	-6	-2	0	1	2	0	0	4	0	-13	-1	-	-1
a15	-2	2	-9	-8	-3	-2	-4	1	-2	-2	-5	-18	-4	1	-

Valuation domain: [-100; +100]

Fig. 8.1: The bipolar-valued pairwise majority margins

It gets easily apparent that candidate $a12$ constitutes a *Condorcet* winner, i.e. the candidate who beats all the other candidates and, with the given voting profile $gavp$, should without doubt win the election. This strongly confirms the first-ranked result obtained with the previous net approval scoring.

Let us eventually compute, with the help of the NetFlows ranking rule), a linear ranking of the 15 eligible candidates and compare the result with the net approval scores' ranking.

```

1 >>> from linearOrders import NetFlowsOrder
2 >>> nf = NetFlowsOrder(m,Comments=True)
3 >>> print('NetFlows versus Net Approval Ranking')
4 >>> print('Candidate\tNetFlows score\tNet Approval score')
5 >>> for item in nf.netFlows:
6 ...     print( '%9s\t  %+.3f\t %+.1f' %\
7 ...             (item[1], item[0], bavp.netApprovalScores[item[1]]) )
8
9 NetFlows versus Net Approval Ranking
10 Candidate      NetFlows score  Net Approval score
11      a12         +410.000         +18.0
12      a03         +142.000          +6.0
13      a04          +98.000          +3.0
14      a05          +54.000          +3.0
15      a11          +34.000          +2.0
16      a09         -16.000          -1.0
17      a14         -20.000          +0.0
18      a13         -22.000          +0.0
19      a06         -50.000          -3.0
20      a07         -74.000          -2.0
21      a02         -96.000          -3.0
22      a08        -102.000          -4.0
23      a15        -110.000          -4.0
24      a10        -122.000          -6.0
25      a01        -126.000          -5.0

```

On the *better approved than* majority margins digraph m , the *NetFlows* rule delivers a ranking that is very similar to the one previously obtained with the corresponding *Net Approval* scores. Only minor inversions do appear, like in the midfield, where candidate $a09$ advances before candidates $a13$ and $a14$ and $a6$ and $a07$ swap their positions 9 and 10. And, the two last-ranked candidates also swap their positions.

This confirms again the pertinence of the net approval scoring approach for finding the winner in a bipolar approving voting system. Yet, voting by approving (+1), disapproving (-1) or ignoring (0) eligible candidates, may also be seen as a performance evaluation of the eligible candidates on a $\{-1, 0, 1\}$ -graded ordinal scale.

8.3 Three-valued evaluative voting system

Following such an epistemic perspective, we may effectively convert the given *BipolarApprovalVotingProfile* instance into a *PerformanceTableau* instance, so as to get access to a corresponding *outranking* decision aiding approach.

Mind that, contrary to the majority margins of the *better approved than* relation, all voters consider now the approved candidates to be all equivalent (+1). Same is true for the disapproved (-1), respectively the ignored candidates (0). The voter's marginal preferences model this time a complete preorder with three equivalence classes.

From the saved file *AVPerfTab.py* (see Line 1 below), we may construct an outranking relation on the eligible candidates with our standard *BipolarOutrankingDigraph* constructor. The semantics of this outranking relation are the following:

- We say that Candidate x **outranks** Candidate y when there is a majority of voters who consider x **at least as well evaluated as** y .(see Line3 below).
- We say that Candidate x **is not outranked by** Candidate y when there is a majority of voters who consider x **not at least as well evaluated as** y .

```

1 >>> bavp.save2PerfTab(fileName='AVPerfTab',valueDigits=0)
2 *--- Saving as performance tableau in file: <AVPerfTab.py> ---*
3 >>> from outrankingDigraphs import BipolarOutrankingDigraph
4 >>> odg = BipolarOutrankingDigraph('AVPerfTab')
5 >>> odg
6 *----- Object instance description -----*
7 Instance class      : BipolarOutrankingDigraph
8 Instance name       : rel_AVPerfTab
9 # Actions           : 15
10 # Criteria          : 100
11 Size                : 210
12 Determinateness (%) : 69.29
13 Valuation domain    : [-1.00;1.00]
14 Attributes          : ['name', 'actions', 'order,
15                        'criteria', 'evaluation', 'NA',
16                        'valuationdomain', 'relation',
17                        'gamma', 'notGamma', ...]
```

The size ($210 = 15 \times 14$) of the resulting outranking digraph *odg*, shown in Line 11 above, reveals that the corresponding *at least as good evaluated as* (outranking) relation models actually a trivial *complete* digraph. All candidates appear to be **equally at least as well evaluated** and the *better evaluated than* (strict outranking) *codual* outranking digraph becomes in fact empty. The converted performance tableau does apparently not contain sufficiently discriminatory performance evaluations for supporting any strict preference situations.

Yet, we may nevertheless try to apply again the *NetFlows* ranking rule to this complete outranking digraph *g* and print side by side the corresponding *NetFlows* scores and the previous *Net Approval* scores.

```

1 >>> from linearOrders import NetFlowsOrder
2 >>> nf = NetFlowsOrder(odg)
3 >>> print('NetFlows versus Net Approval Ranking')
4 >>> print('Candidate\tNetFlows Score\tNet Approval Score')
5 >>> for item in nf.netFlows:
6 ...     print('%9s\t %+.3f\t %+.0f' %\
7 ...           (item[1], item[0],bavp.netApprovalScores[item[1]]))
8
9 NetFlows versus Net Approval Ranking
```

(continues on next page)

(continued from previous page)

Candidate	NetFlows score	Net Approval score
a12	+4.100	+18.0
a03	+1.420	+6.0
a04	+0.980	+3.0
a05	+0.540	+3.0
a11	+0.340	+2.0
a09	-0.160	-1.0
a14	-0.200	+0.0
a13	-0.220	+0.0
a06	-0.500	-3.0
a07	-0.740	-2.0
a02	-0.960	-3.0
a08	-1.020	-4.0
a15	-1.100	-4.0
a10	-1.220	-6.0
a01	-1.260	-5.0

Despite its apparent poor strict preference discriminating power, we obtain here *NetFlows* scores that are directly proportional (divided by 100) to the scores obtained with the *better approved than* majority margins digraph m .

Encouraged by this positive result, we may furthermore try to compute as well a *best choice recommendation*.

```
>>> odg.showBestChoiceRecommendation()
*****
Rubis best choice recommendation(s) (BCR)
(in decreasing order of determinateness)
Credibility domain: [-1.00,1.00]
=== >> ambiguous first choice(s)
* choice          : ['a01', 'a02', 'a03', 'a04', 'a05',
                    'a06', 'a07', 'a08', 'a09', 'a10',
                    'a11', 'a12', 'a13', 'a14', 'a15']
independence      : 0.06
dominance         : 1.00
absorbency        : 1.00
covering (%)      : 100.00
determinateness (%) : 61.13
- most credible action(s) = {
'a12': 0.44, 'a03': 0.34, 'a04': 0.30,
'a14': 0.28, 'a13': 0.24, 'a06': 0.24,
'a11': 0.20, 'a10': 0.20, 'a07': 0.20,
'a01': 0.20, 'a08': 0.18, 'a05': 0.18,
'a15': 0.14, 'a09': 0.14, 'a02': 0.06, }
=== >> ambiguous last choice(s)
* choice          : ['a01', 'a02', 'a03', 'a04', 'a05',
                    'a06', 'a07', 'a08', 'a09', 'a10',
```

(continues on next page)

(continued from previous page)

```
24             'a11', 'a12', 'a13', 'a14', 'a15']
25     independence      : 0.06
26     dominance         : 1.00
27     absorbency        : 1.00
28     covered (%)       : 100.00
29     determinateness (%) : 63.73
30     - most credible action(s) = {
31         'a13': 0.36, 'a06': 0.36, 'a15': 0.34,
32         'a01': 0.34, 'a08': 0.32, 'a07': 0.30,
33         'a02': 0.30, 'a14': 0.28, 'a11': 0.28,
34         'a09': 0.28, 'a04': 0.26, 'a10': 0.24,
35         'a05': 0.20, 'a03': 0.20, 'a12': 0.06, }
```

The outranking digraph *odg* being actually *empty*, we obtain a unique **ambiguous** –first as well as last– choice recommendation which trivially retains all fifteen candidates (see Lines 6-9 above). Yet, the bipolar-valued best choice membership characteristic vector reveals that, among all the fifteen potential winners, it is indeed Candidate *a12* the most credible one with a 72% majority of voters’ support (see Line 16, $(0.44 + 1.0)/2 = 0.72$); followed by Candidate *a03* (67%) and Candidate *a04* (65%). Similarly, Candidates *a13* and *a06* represent the most credible losers with a 68% majority voters’ support (Line 31).

Note: We observe here empirically that **evaluative** voting systems, using three-valued ordinal performance scales, match closely **bipolar approval** voting systems. The latter voting system models, however, more *faithfully* the very preferential information that is expressed with *approved*, *disapproved* or *ignored* statements. The corresponding evaluation on a three-graded scale, being value (numbers) based, cannot express the fact that in bipolar approval voting systems there is **no preferential information** given concerning the pairwise comparison of all *approved*, respectively *disapproved* or *ignored* candidates.

Let us finally illustrate how bipolar approval voting systems may favour multipartisan supported candidates. We shall therefore compare *bipolar approval* versus *uninominal plurality* election results when considering a highly divisive and partisan political context.

8.4 Favouring multipartisan candidates

In modern democracy, politics are largely structured by political parties and activists movements. Let us so consider a bipolar approval voting profile *dvp* where the random voter behaviour is simulated from two pre-electoral polls concerning a political scene with essentially two major competing parties, like the one existing in the US.

```
1 >>> dvp = RandomBipolarApprovalVotingProfile(\
2 ...     numberOfCandidates=15,
3 ...     numberOfVoters=100,
```

(continues on next page)

(continued from previous page)

```
4 ... approvalProbability=0.25,
5 ... disapprovalProbability=0.25,
6 ... WithPolls=True,
7 ... partyRepartition=0.5,
8 ... other=0.05,
9 ... DivisivePolitics=True,
10 ... seed=200)
11
12 >>> dvp.showRandomPolls()
13 Random repartition of voters
14 Party_1 supporters : 45 (45.00%)
15 Party_2 supporters : 49 (49.00%)
16 Other voters      : 6 (06.00%)
17 *----- random polls -----
18 Party_1(45.0%) | Party_2(49.0%) | expected
19 -----
20 a05 : 24.10% | a07 : 24.10% | a07 : 11.87%
21 a14 : 23.48% | a10 : 23.48% | a10 : 11.60%
22 a03 : 15.13% | a01 : 15.13% | a05 : 10.91%
23 a12 : 07.55% | a04 : 07.55% | a14 : 10.67%
24 a08 : 07.11% | a09 : 07.11% | a01 : 07.67%
25 a15 : 04.37% | a13 : 04.37% | a03 : 07.09%
26 a11 : 03.99% | a02 : 03.99% | a04 : 04.55%
27 a06 : 03.80% | a06 : 03.80% | a09 : 04.49%
28 a02 : 02.79% | a11 : 02.79% | a12 : 04.32%
29 a13 : 02.63% | a15 : 02.63% | a08 : 04.30%
30 a09 : 02.24% | a08 : 02.24% | a06 : 03.57%
31 a04 : 01.89% | a12 : 01.89% | a13 : 03.32%
32 a01 : 00.57% | a03 : 00.57% | a15 : 03.25%
33 a10 : 00.20% | a14 : 00.20% | a02 : 03.21%
34 a07 : 00.14% | a05 : 00.14% | a11 : 03.16%
```

The divisive political situation is reflected by the fact that Party_1 and Party_2 supporters show strict reversed preferences. The leading candidates of Party_1 (*a05* and *a14*) are last choices for Party_2 supporters and, Candidates *a07* and *a10*, leading candidates for Party_2 supporters, are similarly the least choices for Party_1 supporters.

No clear winner may be guessed from these pre-election polls. As Party_2 shows however slightly more supporters than Party_1, the expected winner in an uninominal *plurality* or *instant-runoff* voting system will be Candidate *a07*, i.e, the leading candidate of Party_2 (see below).

```
1 >>> dvp.computeSimpleMajorityWinner()
2 ['a07']
3 >>> dvp.computeInstantRunoffWinner()
4 ['a07']
```

Now, in a corresponding bipolar approval voting system, Party_1 supporters will usu-

ally approve their leading candidates and disapprove the leading candidates of Party_2. Vice versa, Party_2 supporters will usually approve their leading candidates and disapprove the leading candidates of Party_1. Let us consult the resulting approval votes per candidate.

```

1 >>> dvp.showApprovalResults()
2   Candidate: a07 obtains 30 votes
3   Candidate: a10 obtains 28 votes
4   Candidate: a05 obtains 28 votes
5   Candidate: a01 obtains 28 votes
6   Candidate: a03 obtains 26 votes
7   Candidate: a02 obtains 26 votes
8   Candidate: a12 obtains 25 votes
9   Candidate: a14 obtains 24 votes
10  Candidate: a13 obtains 24 votes
11  Candidate: a09 obtains 21 votes
12  Candidate: a04 obtains 21 votes
13  Candidate: a08 obtains 19 votes
14  Candidate: a06 obtains 17 votes
15  Candidate: a15 obtains 15 votes
16  Candidate: a11 obtains 12 votes
17  Total approval votes: 344
18  Approval proportion: 344/1500 = 0.23

```

When considering only the approval votes, we find confirmed above that the leading candidate of Party_2 obtains in this simulation a plurality of approval votes. In uninominal *plurality* or *instant-runoff* voting systems, this candidate wins hence the election, quite to the despair of Party_1 supporters. As a foreseeable consequence, this election result will be more or less aggressively contested which leads to a loss of popular trust in democratic elections and institutions.

If we look however on the corresponding disapprovals, we discover that, not surprisingly, the leading candidates of both parties collect by far the highest number of disapproval votes.

```

1 >>> dvp.showDisapprovalResults()
2   Candidate: a02 obtains 14 votes
3   Candidate: a04 obtains 14 votes
4   Candidate: a13 obtains 14 votes
5   Candidate: a06 obtains 15 votes
6   Candidate: a09 obtains 15 votes
7   Candidate: a08 obtains 16 votes
8   Candidate: a11 obtains 16 votes
9   Candidate: a15 obtains 18 votes
10  Candidate: a12 obtains 20 votes
11  Candidate: a01 obtains 29 votes
12  Candidate: a03 obtains 30 votes
13  Candidate: a10 obtains 37 votes

```

(continues on next page)

(continued from previous page)

```
14 Candidate: a07 obtains 44 votes
15 Candidate: a14 obtains 45 votes
16 Candidate: a05 obtains 49 votes
17 Total disapproval votes: 376
18 Disapproval proportion: 376/1500 = 0.25
```

Balancing now approval against disapproval votes will favour the moderate, bipartisan supported, candidates.

```
1 >>> dvp.showNetApprovalScores()
2 Net Approval Scores
3 Candidate: a02 obtains 12 net approvals
4 Candidate: a13 obtains 10 net approvals
5 Candidate: a04 obtains 7 net approvals
6 Candidate: a09 obtains 6 net approvals
7 Candidate: a12 obtains 5 net approvals
8 Candidate: a08 obtains 3 net approvals
9 Candidate: a06 obtains 2 net approvals
10 Candidate: a01 obtains -1 net approvals
11 Candidate: a15 obtains -3 net approvals
12 Candidate: a11 obtains -4 net approvals
13 Candidate: a03 obtains -4 net approvals
14 Candidate: a10 obtains -9 net approvals
15 Candidate: a07 obtains -14 net approvals
16 Candidate: a14 obtains -21 net approvals
17 Candidate: a05 obtains -21 net approvals
```

Candidate *a02*, appearing in the pre-electoral polls in the midfield (in position 7 for Party_2 and in position 9 for Party_1 supporters), shows indeed the highest net approval score. Second highest net approval score obtains Candidate *a13*, in position 6 for Party_2 and in position 10 for Party_1 supporters.

Fig. 8.2, showing the *NetFlows* ranked relation table of the *better approved than* majority margins digraph, confirms below this net approval scoring result.

```
>>> m = MajorityMarginsDigraph(dvp)
>>> m.showHTMLRelationTable(\
...     actionsList=m.computeNetFlowsRanking(),
...     relationName='r(x > y)')
```

r(x > y)	a02	a13	a04	a09	a12	a08	a06	a01	a11	a15	a03	a10	a07	a14	a05
a02	-	6	5	6	9	5	10	12	12	14	11	15	22	22	23
a13	-6	-	2	5	2	5	8	10	14	10	9	13	18	23	20
a04	-5	-2	-	0	2	2	3	7	7	8	11	13	18	21	18
a09	-6	-5	0	-	0	2	5	5	11	9	5	13	16	21	16
a12	-9	-2	-2	0	-	4	6	2	9	6	10	5	10	23	25
a08	-5	-5	-2	-2	-4	-	4	0	8	9	7	5	11	21	20
a06	-10	-8	-3	-5	-6	-4	-	-2	5	5	5	6	13	17	18
a01	-12	-10	-7	-5	-2	0	2	-	1	-1	3	8	11	9	13
a11	-12	-14	-7	-11	-9	-8	-5	-1	-	1	-2	7	14	13	16
a15	-14	-10	-8	-9	-6	-9	-5	1	-1	-	0	3	10	14	14
a03	-11	-9	-11	-5	-10	-7	-5	-3	2	0	-	-3	7	16	16
a10	-15	-13	-13	-13	-5	-5	-6	-8	-7	-3	3	-	3	6	7
a07	-22	-18	-18	-16	-10	-11	-13	-11	-14	-10	-7	-3	-	3	5
a14	-22	-23	-21	-21	-23	-21	-17	-9	-13	-14	-16	-6	-3	-	0
a05	-23	-20	-18	-16	-25	-20	-18	-13	-16	-14	-16	-7	-5	0	-

Valuation domain: [-100; +100]

Fig. 8.2: The pairwise *better approved than* majority margins

Candidate *a02* appears indeed *better approved than* any other candidate (*Condorcet* winner); and, the leading candidates of Party_1, *a05* and *a14*, are *less approved than* any other candidates (weak *Condorcet* losers).

```

1 >>> m.computeCondorcetWinners()
2 ['a02']
3 >>> m.computeWeakCondorcetLosers()
4 ['a05', 'a14']

```

We see this result furthermore confirmed when computing the corresponding **first**, respectively **last** choice recommendation.

```

1 >>> m.showBestChoiceRecommendation()
2 Rubis best choice recommendation(s) (BCR)
3 (in decreasing order of determinateness)
4 Credibility domain: [-100.00,100.00]
5 == >> potential first choice(s)
6 * choice : ['a02']
7 independence : 100.00
8 dominance : 5.00
9 absorbency : -23.00
10 covering (%) : 100.00
11 determinateness (%) : 52.50
12 - most credible action(s) = { 'a02': 5.00, }

```

(continues on next page)

(continued from previous page)

```
13  === >> potential last choice(s)
14  * choice      : ['a05', 'a14']
15  independence  : 0.00
16  dominance     : -23.00
17  absorbency    : 5.00
18  covered (%)   : 100.00
19  determinateness (%) : 50.00
20  - most credible action(s) = { }
```

Candidate *a02*, being actually a *Condorcet* winner, gives an initial **dominating kernel** of digraph *m*, whereas Party_1 leading Candidates *a05* and *a14*, both being weak *Condorcet* losers, give together a terminal **dominated** prekernel. They hence represent our **first choice**, respectively, **last choice** recommendations for winning this simulated election.

Let us conclude by predicting that, for leading political candidates in an aggressively divisive political context, the perspective to easily fail election with bipolar approval voting systems, might or will induce a change in the usual way of running electoral campaigns. Political parties and politicians, who avoid aggressive competitive propaganda and instead propose multipartisan collaborative social choices, will be rewarded with better election results than any kind of extremism. It could mean the end of sterile political obstructions and war like electoral battles.

Let's do it.

Note: It is worthwhile noticing the essential structural and computational role, the **zero value** is again playing in bipolar approval voting systems. This epistemic and logical **neutral** term is needed indeed for handling in a consistent and efficient manner **not communicated votes** and/or **indeterminate preferential statements**.

Back to [Content Table](#) (page 2)

9 Bibliography

References

- [BIS-2015p] Bisdorff R. (2015). “The EURO 2004 Best Poster Award: Choosing the Best Poster in a Scientific Conference”. Chapter 5 in R. Bisdorff, L. Dias, P. Meyer, V. Mousseau, and M. Pirlot (Eds.), *Evaluation and Decision Models with Multiple Criteria: Case Studies*. Springer-Verlag Berlin Heidelberg, International Handbooks on Information Systems, DOI 10.1007/978-3-662-46816-6_1, pp. 117-166 (downloadable [PDF file 754.7 kB](#) (<http://hdl.handle.net/10993/23714>)).

- [BIS-2014p] Bisdorff R., Meyer P. and Veneziano Th. (2014). “Elicitation of criteria weights maximising the stability of pairwise outranking statements”. *Journal of Multi-Criteria Decision Analysis* (Wiley) 21: 113-124 (downloadable preprint PDF file 431.4 Kb (<http://hdl.handle.net/10993/23701>)).
- [BIS-2013p] Bisdorff R. (2013) “On Polarizing Outranking Relations with Large Performance Differences” *Journal of Multi-Criteria Decision Analysis* (Wiley) 20:3-12 (downloadable preprint PDF file 403.5 Kb (<http://hdl.handle.net/10993/245>)).
- [BAU-2013p] Baujard A., Gavrel F., Igersheim H., Laslier J.-F. and Lebon I. (2013). “Approval Voting, Evaluation Voting: An Experiment during the 2012 French Presidential Election”. In *Revue Économique* (Presses de Sciences Po) Volume 64, Issue 2, pp. 345-356 (downloadable English translation, 652.5 kB).
- [BIS-2012p] Bisdorff R. (2012). “On measuring and testing the ordinal correlation between bipolar outranking relations”. In *Proceedings of DA2PL’2012 From Multiple Criteria Decision Aid to Preference Learning*, University of Mons 91-100. (downloadable preliminary version PDF file 408.5 kB (<http://hdl.handle.net/10993/23909>)).
- [BIS-2006_1p] Bisdorff R., Pirlot M. and Roubens M. (2006). “Choices and kernels from bipolar valued digraphs”. *European Journal of Operational Research*, 175 (2006) 155-170. (Online) Electronic version: DOI:10.1016/j.ejor.2005.05.004 (downloadable preliminary version PDF file 257.3Kb (<http://hdl.handle.net/10993/23720>)).
- [BIS-2006_2p] Bisdorff R. (2006). “On enumerating the kernels in a bipolar-valued digraph”. *Annales du Lamsade* 6, Octobre 2006, pp. 1 - 38. Université Paris-Dauphine. ISSN 1762-455X (downloadable version PDF file 532.2 Kb (<http://hdl.handle.net/10993/38741>)).
- [BIS-2004_1p] Bisdorff R. (2004). “Concordant Outranking with multiple criteria of ordinal significance”. 4OR, *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, Springer-Verlag, Issue: Volume 2, Number 4, December 2004, Pages: 293 - 308. [ISSN: 1619-4500 (Paper) 1614-2411 (Online)] Electronic version: DOI: 10.1007/s10288-004-0053-7 (downloadable preliminary version PDF file 137.1Kb (<http://hdl.handle.net/10993/23721>)).
- [BIS-2004_2p] Bisdorff R. (2004). Preference aggregation with multiple criteria of ordinal significance. In: D. Bouyssou, M. Janowitz, F. Roberts, and A. Tsouki’s (eds.), *Annales du LAMSADE*, 3, Octobre 2004, Université Paris-Dauphine, pp. 25-44 [ISSN 1762-455X] (downloadable PDF file 167.6Kb (<http://hdl.handle.net/10993/42420>)).
- [SCH-1985p] Schmidt G. and Ströhlein Th. (1985), “On kernels of graphs and solutions of games: a synopsis based on relations and fixpoints”. SIAM, *J. Algebraic Discrete Methods*, 6, 1985, 54–65.
- [KEN-1938p] Kendall M.G. (1938), “A New Measure of Rank Correlation”. *Biometrika* 30:81–93