Multi-Criteria Decision Support using ELECTRE methods

Roman Słowiński

Poznań University of Technology, Poland

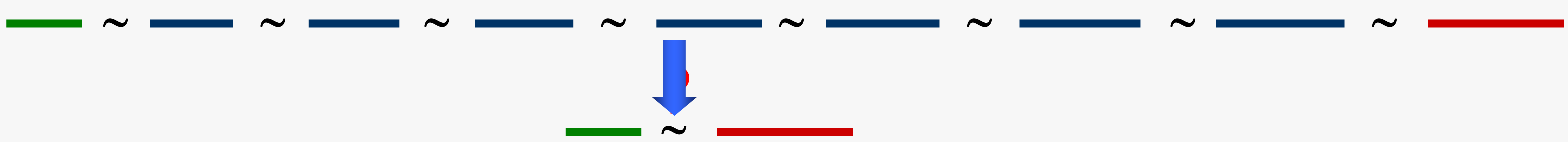
Weak points of preference modelling using utility (value) function

- Utility function distinguishes only 2 possible relations between actions:

  preference relation: $a \succ b \Leftrightarrow U(a) > U(b)$

  indifference relation: $a \sim b \Leftrightarrow U(a) = U(b)$

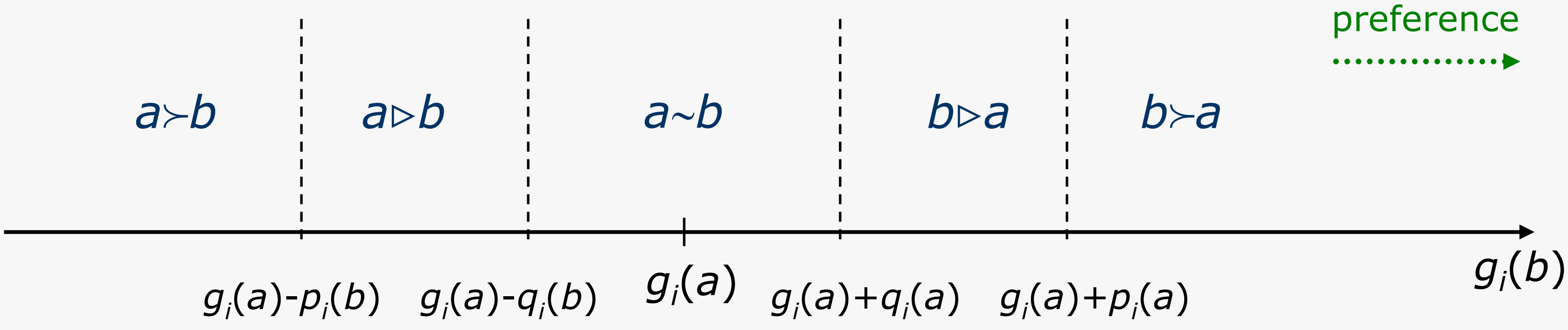
- $\succ$ is asymmetric (antisymmetric and irreflexive) and transitive
- $\sim$ is symmetric, reflexive and transitive
- Transitivity of indifference is troublesome, e.g.



- In consequence, a non-zero indifference threshold q_i is necessary
- An immediate transition from indifference to preference is unrealistic, so a preference threshold $p_i \geq q_i$ and a weak preference relation $\triangleright$ are desirable
- Another realistic situation which is not modelled by U is incomparability, so a good model should include also an incomparability relation ?

Four basic preference relations and an outranking relation S

- Four basic preference relations are: $\{\sim, \triangleright, \succ, ?\}$



- Criterion with thresholds $p_i(a) \geq q_i(a) \geq 0$ is called pseudo-criterion

- The four basic situations of indifference, strict preference, weak preference, and incomparability are sufficient for establishing a realistic model of Decision Maker's (DM's) preferences

Four basic preference relations and an outranking relation *S*

- Axiom of limited comparability (Roy 1985):

  Whatever the actions considered, the criteria used to compare them, and the information available, one can develop a satisfactory model of DM's preferences by assigning one, or a grouping of two or three of the four basic situations, to any pair of actions.

- Outranking relation *S* groups three basic preference relations:

$$S = \{\sim, \triangleright, \succ\} \text{ – reflexive and non-transitive}$$

aSb means: „action *a* is at least as good as action *b*"

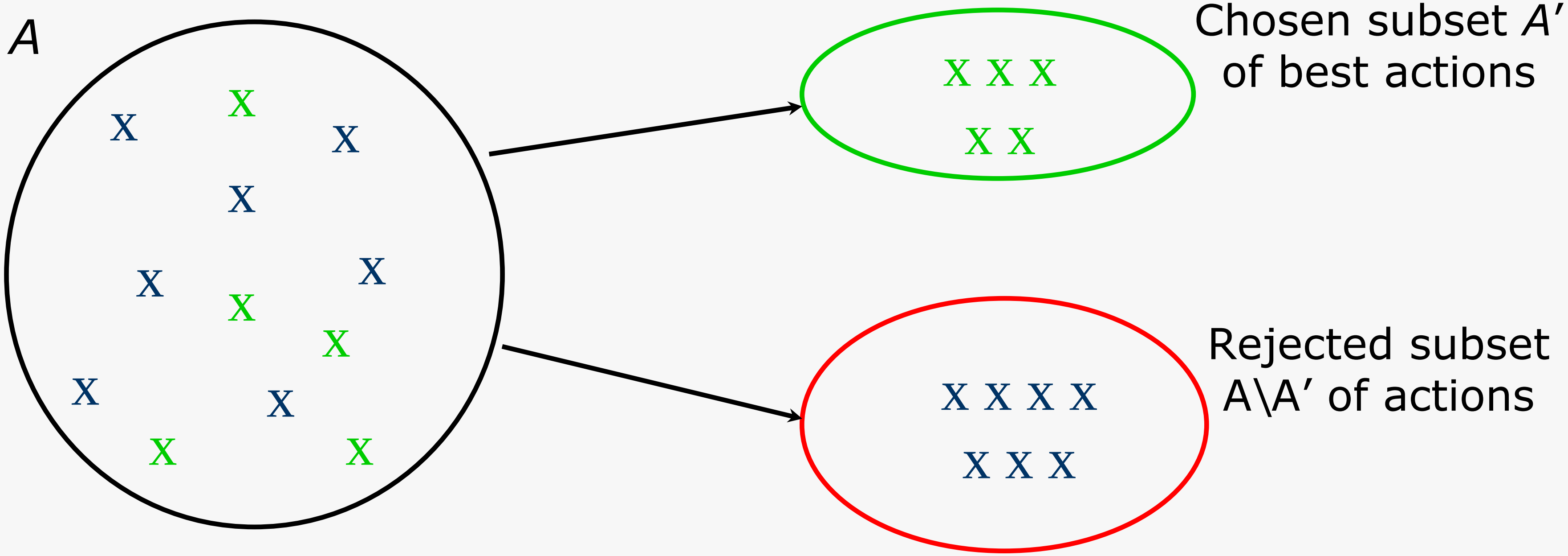
- For each couple $a,b \in A$:

$$aSb \wedge \text{non } bSa \Leftrightarrow a \triangleright b \vee a \succ b$$

$$aSb \wedge bSa \Leftrightarrow a \sim b$$

$$\text{non } aSb \wedge \text{non } bSa \Leftrightarrow a?b$$

ELECTRE Is: choice problem (P_α)



ELECTRE Is

- Input data: finite set of actions $A=\{a, b, c, \ldots, h\}$
  consistent family of criteria $G=\{g_1, g_2, \ldots, g_n\}$

- Preferential information ($i=1,\ldots,k$)

  intracriteria: – indifference thresholds $q_i(a)=\alpha_i^q \times g_i(a)+\beta_i^q$
  – preference thresholds $p_i(a)=\alpha_i^p \times g_i(a)+\beta_i^p$

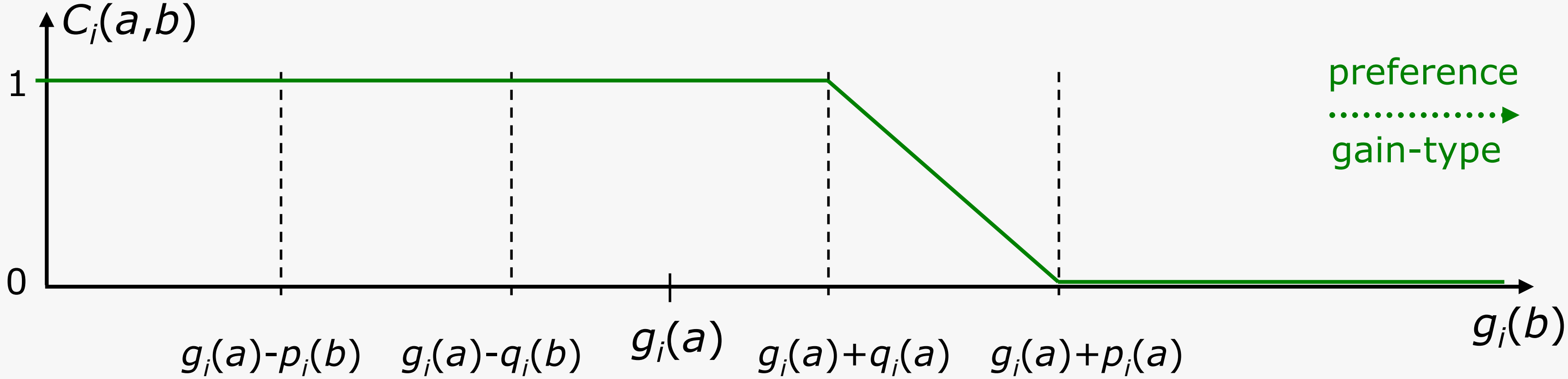
  intercriteria: – importance coefficients (weights) of criteria k_i
  – veto thresholds $v_i(a)=\alpha_i^v \times g_i(a)+\beta_i^v$

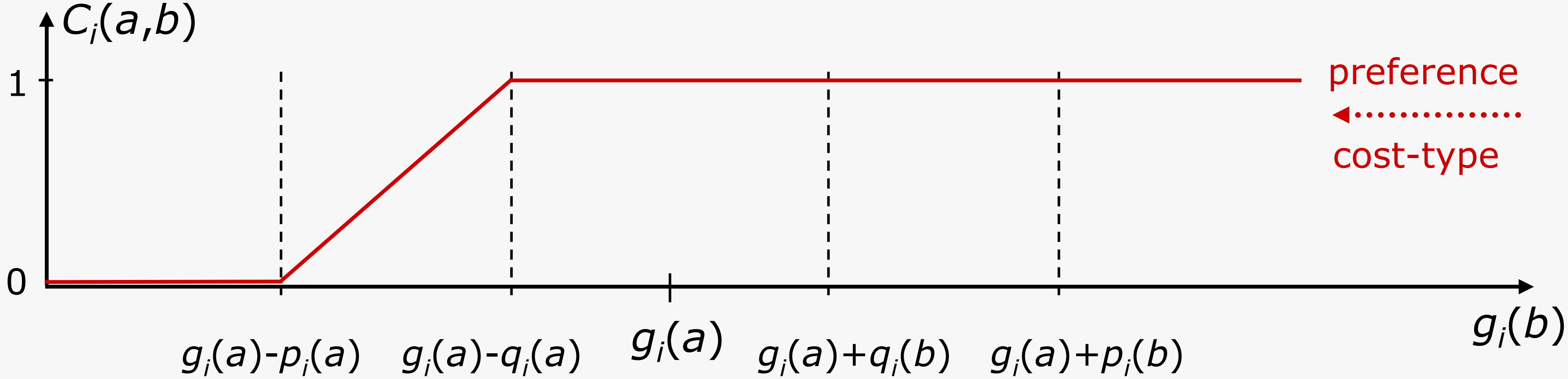
- $0 \leq q_i(a) \leq p_i(a) \leq v_i(a)$ are functions of a worse evaluation of the two being compared

- The preference model, i.e. outranking relation S in set A is constructed via concordance and discordance tests

ELECTRE Is - concordance test

- Checks if the coalition of criteria concordant with the hypothesis *aSb* is strong enough (for each couple $a,b \in A$)
- Concordance coefficient $C_i(a,b)$ for each criterion g_i:





ELECTRE Is - concordance test

- Aggregation of concordance coefficients for $a,b \in A$:

$$C(a,b) = \frac{\sum_{i=1}^{n} k_i C_i(a,b)}{\sum_{i=1}^{n} k_i}$$

- $C(a,b) \in [0, 1]$
- Concordance test is positif if: $C(a,b) \geq \lambda$

  where λ is a cutting level, such that $0.5 \leq \lambda \leq 1 - \min\limits_{i=1,\ldots,n}\{k_i\} \Big/ \sum_{i=1}^{n} k_i$

ELECTRE Is - discordance test and outranking relation

- Checks if among criteria discordant with the hypothesis *aSb* there is a strong opposition against *aSb* (for each $a,b \in A$ such that $C(a,b) \geq \lambda$)

- For a discordant criterion g_i, the opposition is strong if:

  (for gain-type criterion) $g_i(b) - g_i(a) \geq v_i(a)$

  (for cost-type criterion) $g_i(a) - g_i(b) \geq v_i(a)$

- Conclusion: *aSb* is true if and only if $C(a,b) \geq \lambda$ and there is no criterion strongly opposed to the hypothesis

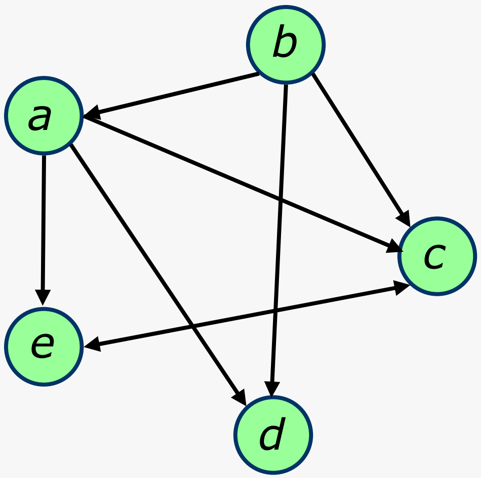
- In result, for each couple $a,b \in A$, one obtains relation S either true (1) or false (0)

- The preference structure in set A can be represented by a graph where nodes represent actions and arcs represent relation S

ELECTRE Is – exploitation of the outranking graph

- The outranking graph *S* represents a preference structure in *A*

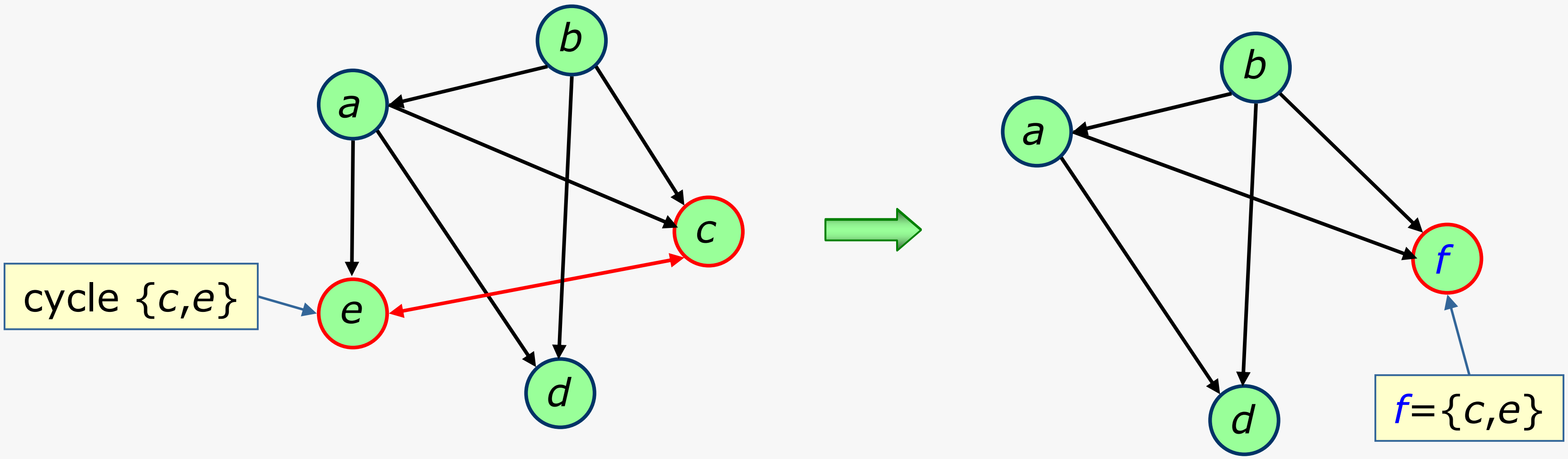
S	*a*	*b*	*c*	*d*	*e*
a	1	0	1	1	1
b	1	1	1	1	0
c	0	0	1	0	1
d	0	0	0	1	0
e	0	0	1	0	1



- What are the best actions ?
- Kernel *K* of the outranking graph:
  - actions (nodes) belonging to *K* do not outrank each other (no arc between them)
  - each action not belonging to *K* is directly outranked by at least one action from *K*

ELECTRE Is – search for the kernel

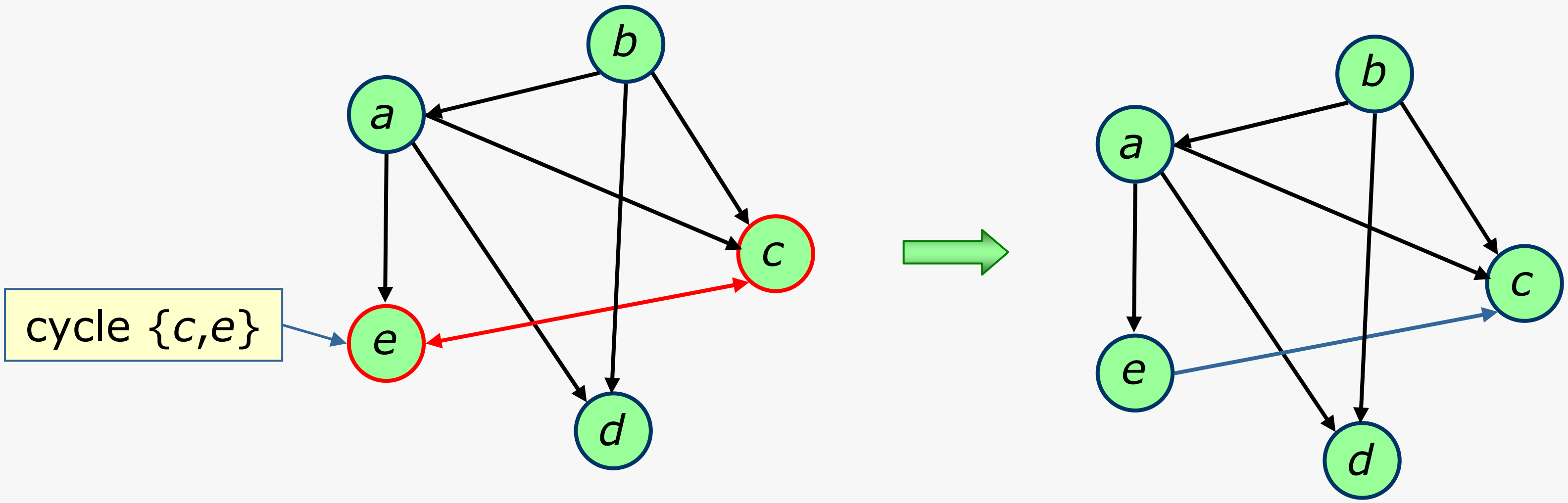
- In order to have a single kernel, the graph should be acyclic



- Cycles should be first eliminated in one of two ways:
  - replacing the cycle by a dummy node representing an equivalence class (clique)

ELECTRE Is – search for the kernel

- In order to have a single kernel, the graph should be acyclic



- Cycles should be first eliminated in one of two ways:
  - replacing the cycle by a dummy node representing an equivalence class (clique)
  - cutting the cycle = eliminationg the „weakest” arcs (outranking relations)

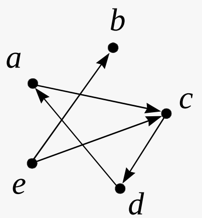
Algorytm wykrywania cykli w grafie

α. Utworzyć tablicę z jednym wejściem. W każdym wierszu wpisać indeksy bezpośrednich następników danego wierzchołka x_i (tzw. listę ZA).

β. Poszukać linię pustą i skreślić numer tej linii wszędzie tam gdzie występuje w tablicy. Linie z samymi skreślonymi indeksami traktować jak puste. Skreślanie kontynuować aż do wyczerpania możliwości.

γ. Jeśli wszystkie indeksy mogły być skreślone, to graf nie ma cykli. W przeciwnym razie ma co najmniej jednej cykl.

δ. Wybrać indeks dowolnej linii niepustej a w niej dowolny nie skreślony indeks i skreślić go. Skreślanie kontynuować do momentu, gdy sekwencja skreślonych indeksów utworzy cykl.



α.

a	*c*
b	-
c	*d*
d	*a*
e	*b*, *c*

β.

a	*c*
~~*b*~~	~~-~~
c	*d*
d	*a*
e	~~*b*~~, *c*

γ.

nie wszystkie indeksy mogły być skreślone

δ.

wybieramy wiersz *a* i indeks *c*

a	~~*c*~~
b	-
c	*d*
d	*a*
e	*b*, *c*

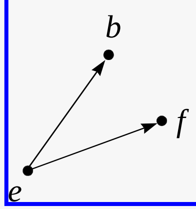
~~*a*~~	~~*c*~~
~~*b*~~	~~-~~
c	*d*
d	*a*
e	*b*, *c*

~~*a*~~	~~*c*~~
~~*b*~~	~~-~~
c	*d*
d	~~*a*~~
e	*b*, *c*

itd. aż do

~~*a*~~	~~*c*~~
~~*b*~~	~~-~~
~~*c*~~	~~*d*~~
~~*d*~~	~~*a*~~
e	*b*, *c*

cykl tworzą: *c*, *d* i *a*



otrzymany graf jeszcze raz prze-puszczamy przez algorytm

α.

b	-
e	*b*, *f*
f	-

β.

~~*b*~~	~~-~~
e	*b*, *f*
f	-

~~*b*~~	~~-~~
e	~~*b*~~, *f*
f	-

~~*b*~~	~~-~~
e	~~*b*~~, *f*
~~*f*~~	~~-~~

~~*b*~~	~~-~~
e	~~*b*~~, ~~*f*~~
~~*f*~~	~~-~~

~~*b*~~	~~-~~
~~*e*~~	~~*b*, *f*~~
~~*f*~~	~~-~~

γ.

wszystkie indeksy mogły być skreślone więc graf jest acy-kliczny

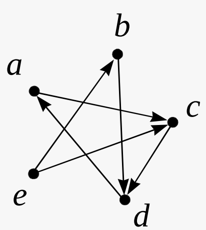
Algorytm wykrywania cykli w grafie

α. Utworzyć tablicę z jednym wejściem. W każdym wierszu wpisać indeksy bezpośrednich **następników** danego wierzchołka x_i (tzw. **listę ZA**).

β. Poszukać linię pustą i skreślić numer tej linii wszędzie tam gdzie występuje w tablicy. Linie z samymi skreślonymi indeksami traktować jak puste. Skreślanie kontynuować aż do wyczerpania możliwości.

γ. Jeśli wszystkie indeksy mogły być skreślone, to graf nie ma cykli. W przeciwnym razie ma co najmniej jednej cykl.

δ. Wybrać indeks dowolnej linii niepustej a w niej dowolny nie skreślony indeks i skreślić go. Skreślanie kontynuować do momentu, gdy sekwencja skreślonych indeksów utworzy **cykl**.



α.

a	*c*
b	*d*
c	*d*
d	*a*
e	*b, c*

β. nie ma pustych linii więc przechodzimy do ostatniego kroku

γ.

δ. wybieramy wiersz *a* i indeks *c*

a	~~*c*~~
b	*d*
c	*d*
d	*a*
e	*b, c*

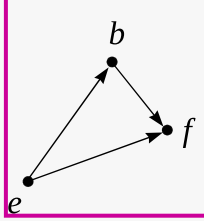
~~*a*~~	~~*c*~~
b	*d*
c	*d*
d	*a*
e	*b, c*

~~*a*~~	~~*c*~~
b	*d*
c	*d*
d	~~*a*~~
e	*b, c*

itd. aż do

~~*a*~~	~~*c*~~
b	*d*
~~*c*~~	~~*d*~~
~~*d*~~	~~*a*~~
e	*b, c*

cykl tworzą: *c*, *d* i *a*



otrzymany graf jeszcze raz przepuszczamy przez algorytm

α.

b	*f*
e	*b, f*
f	-

β.

b	*f*
e	*b, f*
~~*f*~~	~~-~~

b	~~*f*~~
e	*b,* ~~*f*~~
~~*f*~~	~~-~~

~~*b*~~	~~*f*~~
e	*b,* ~~*f*~~
~~*f*~~	~~-~~

~~*b*~~	~~*f*~~
e	~~*b*~~, ~~*f*~~
~~*f*~~	~~-~~

~~*b*~~	~~*f*~~
~~*e*~~	~~*b*~~, ~~*f*~~
~~*f*~~	~~-~~

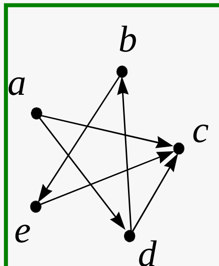
γ. wszystkie indeksy mogły być skreślone więc graf jest acykliczny

Algorytm znajdowania jądra w grafie acyklicznym

α. Utworzyć tablicę z jednym wejściem. W każdym wierszu wpisać indeksy bezpośrednich poprzedników danego wierzchołka x_i (tzw. listę PRZED).

β. Zaznaczyć linię pustą (znaczkiem $\times$). Skreślić całe linie, w których występuje wierzchołek z indeksem zaznaczonej linii pustej.
Skreślić indeks skreślonego wiersza gdziekolwiek pojawia się w tablicy.
Każda linia z samymi skreślonymi indeksami jest uznawana z pustą.
Iterować aż do wyczerpania możliwości.

γ. Jeśli wszystkie indeksy mogły być zaznaczone lub skreślone, to graf ma pojedyncze jądro złożone z wierzchołków odpowiadających zaznaczonym ($\times$) wierszom.



α.

a	-
b	d
c	a, d, e
d	a
e	b

β.

$\times$	a	-
	b	d
	c	a, d, e
	d	a
	e	b

$\times$	a	-
	b	d
	~~c~~	~~a, d, e~~
	~~d~~	~~a~~
	e	b

$\times$	a	-
	b	~~d~~
	~~c~~	~~a, d, e~~
	~~d~~	~~a~~
	e	b

$\times$	a	-
$\times$	b	~~d~~
	~~c~~	~~a, d, e~~
	~~d~~	~~a~~
	e	b

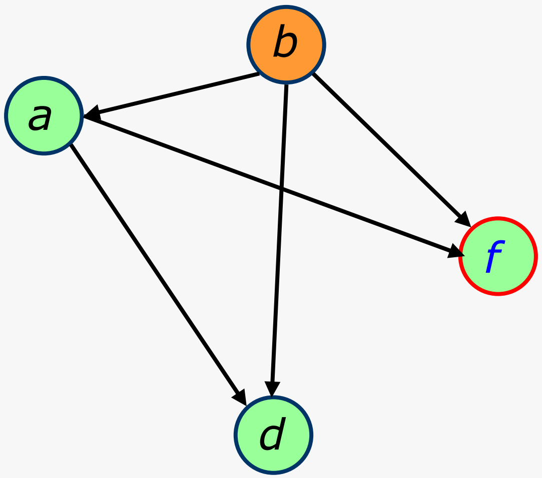
$\times$	a	-
$\times$	b	~~d~~
	~~c~~	~~a, d, e~~
	~~d~~	~~a~~
	~~e~~	~~b~~

γ.

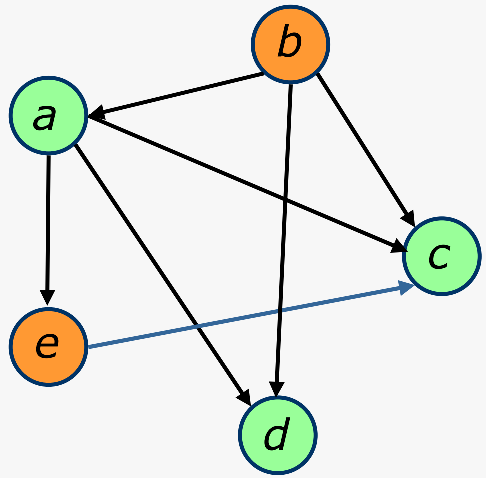
wszystkie wiersze mogły zostać zaznaczone lub skreślone, więc graf ma pojedyncze jądro: $N = \{a, b\}$

ELECTRE Is – final recommendation

- The best actions = the kernel:



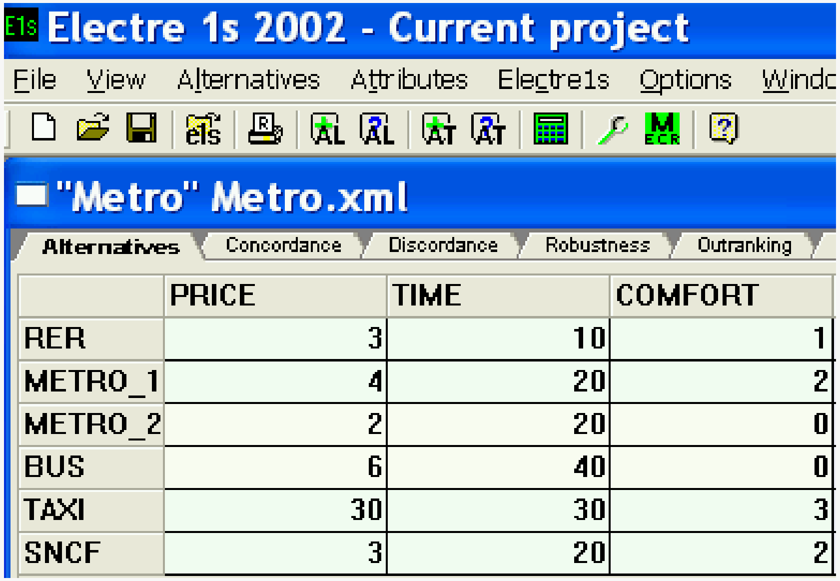
$K=\{b\}$



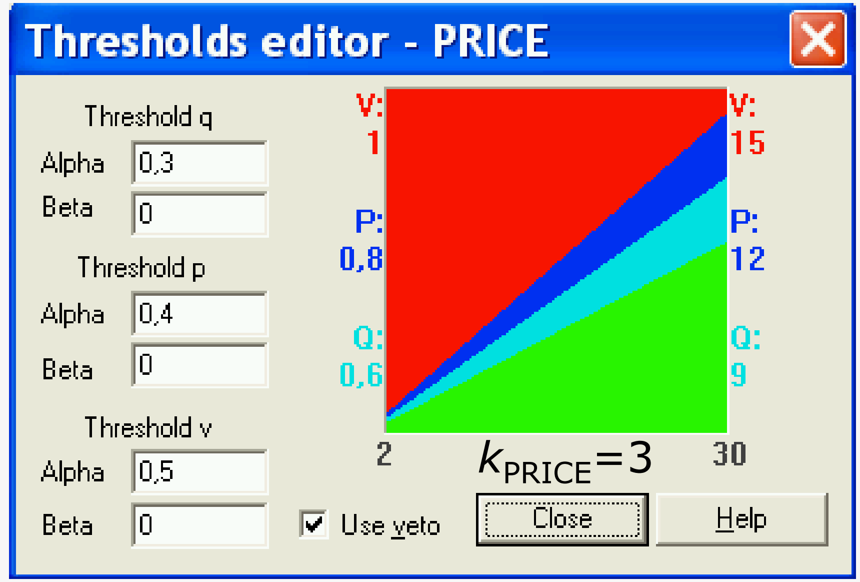
$K=\{b,e\}$

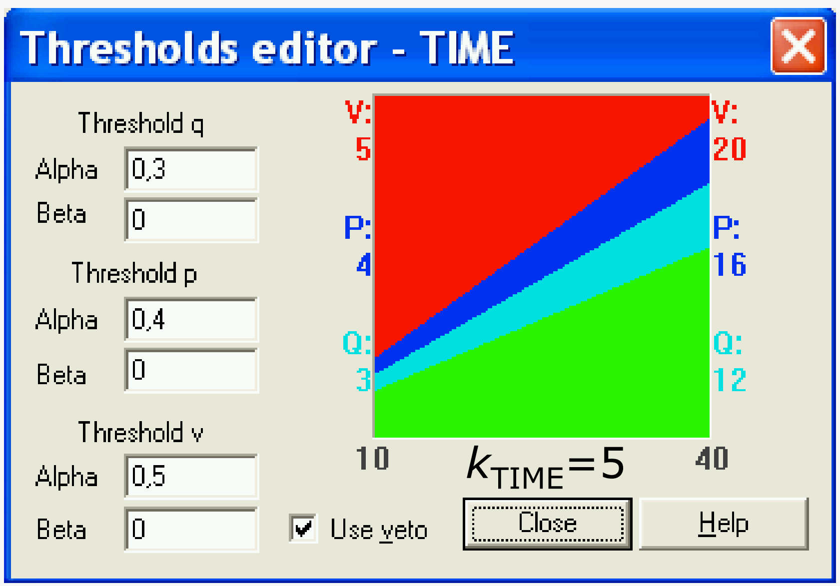
- Actions in the kernel can be:
  - initial (without predecessors)
  - isolated (incomparable to all others)
  - intermediate (both outranking and outranked)
  - final (without successors)

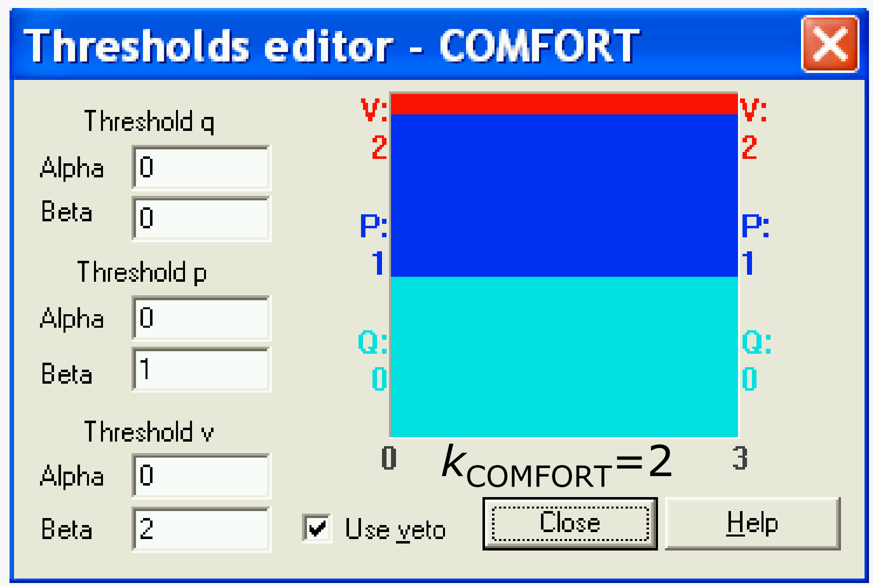
ELECTRE Is - example



	PRICE	TIME	COMFORT
RER	3	10	1
METRO_1	4	20	2
METRO_2	2	20	0
BUS	6	40	0
TAXI	30	30	3
SNCF	3	20	2







ELECTRE Is - example

Alternatives | **Concordance** | Discordance | Robustness | Outranking | Outranking Graph | Final Graph

	RER	METRO_1	METRO_2	BUS	TAXI	SNCF
RER	1	0,8	0,9	1	0,8	0,8
METRO_1	0,5	1	0,7	1	0,8	1
METRO_2	0,3	0,8	1	1	0,8	0,8
BUS	0	0,2	0,2	1	0,8	0
TAXI	0,2	0,53333333333	0,53333333333	0,7	1	0,53333333333
SNCF	0,5	1	0,9	1	0,8	1

$\lambda=0.75$

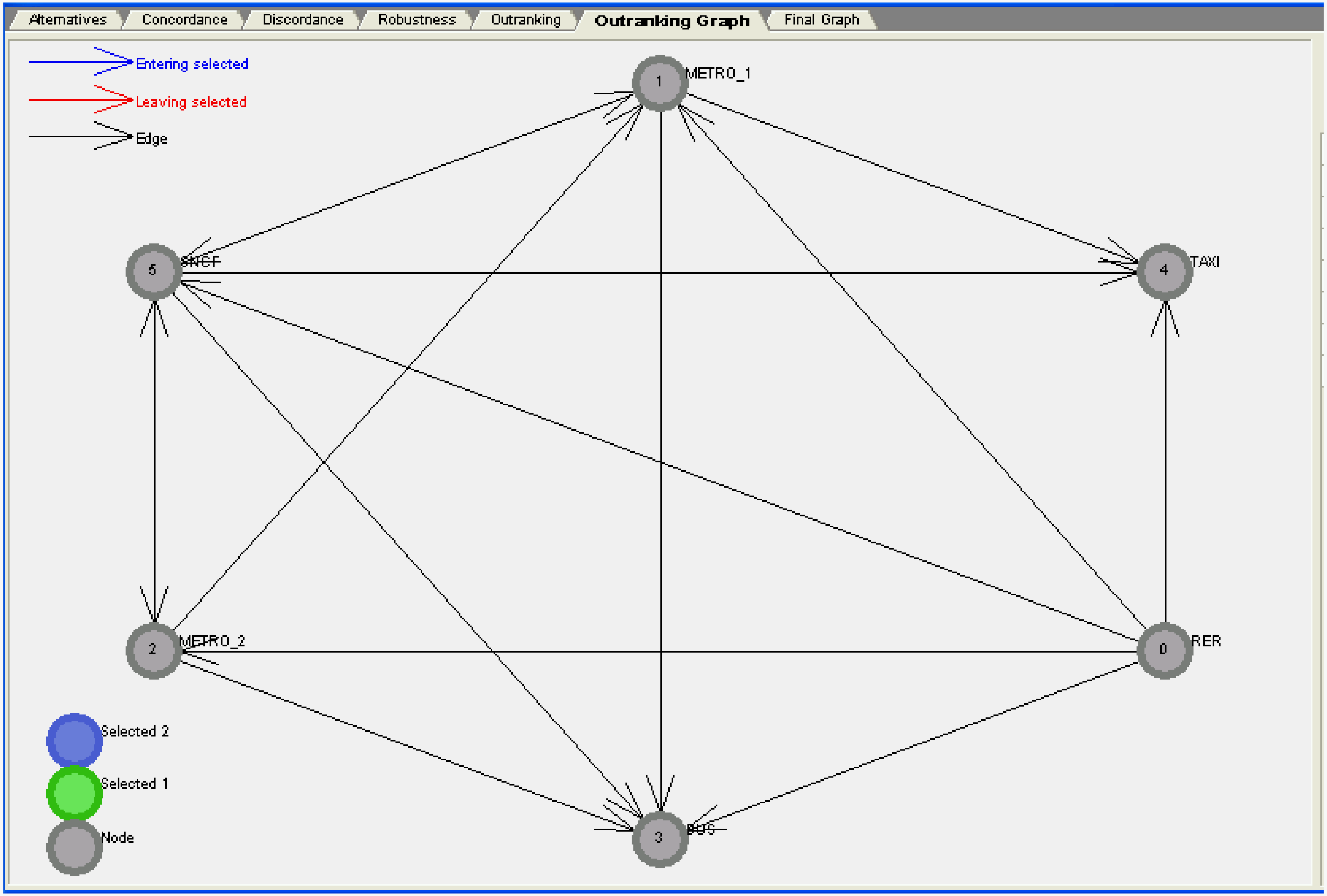
Alternatives | Concordance | **Discordance** | Robustness | Outranking | Outranking Graph | Final Graph

	RER	METRO_1	METRO_2	BUS	TAXI	SNCF
RER	0	0	0	0	0	0
METRO_1	-	0	-	0	0	0
METRO_2	-	0	0	0	1	0
BUS	-	-	-	0	1	-
TAXI	-	-	-	-	0	-
SNCF	-	0	0	0	0	0

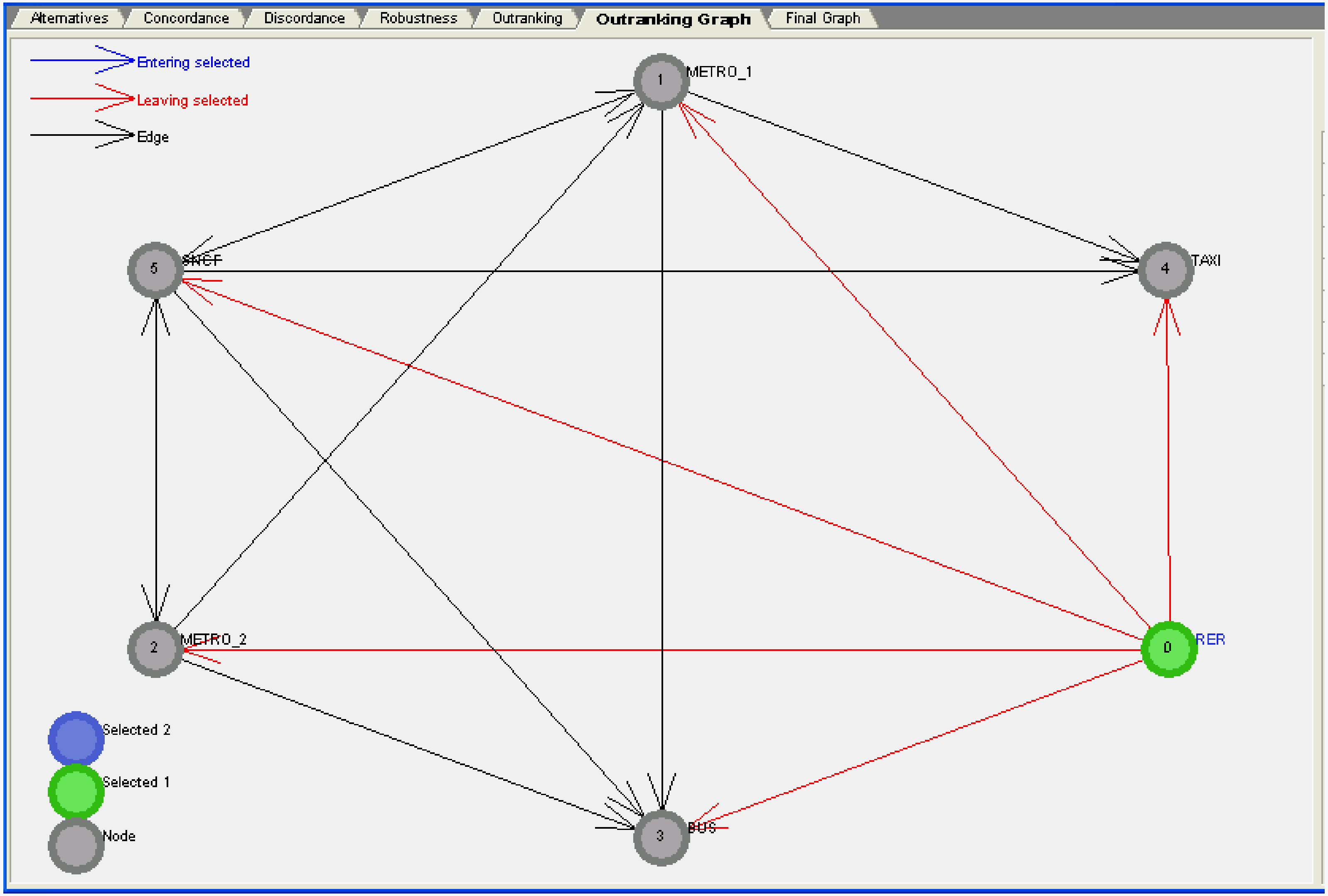
Alternatives | Concordance | Discordance | Robustness | **Outranking** | Outranking Graph | Final Graph

	RER	METRO_1	METRO_2	BUS	TAXI	SNCF
RER	1	1	1	1	1	1
METRO_1	0	1	0	1	1	1
METRO_2	0	1	1	1	0	1
BUS	0	0	0	1	0	0
TAXI	0	0	0	0	1	0
SNCF	0	1	1	1	1	1

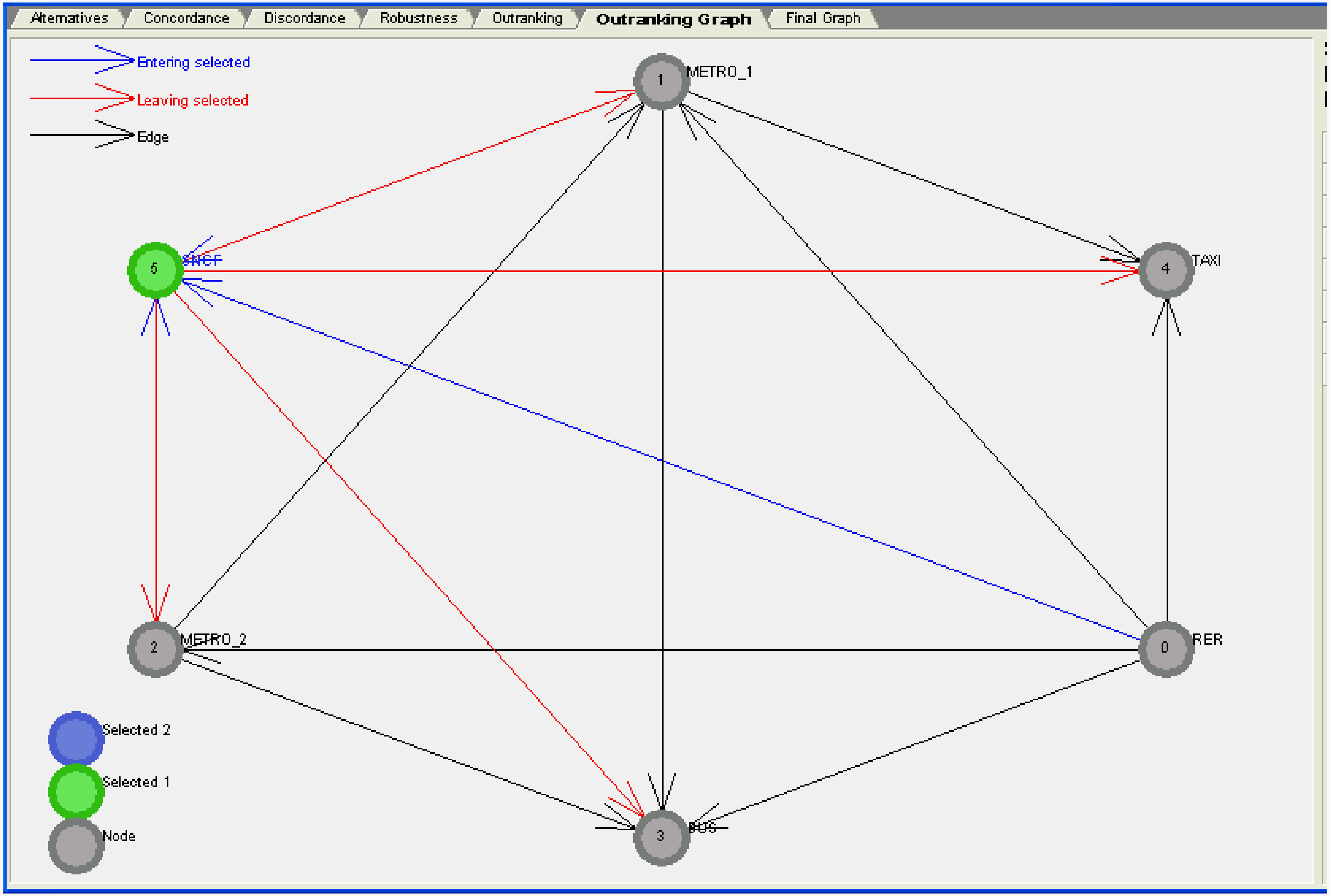
ELECTRE Is - example



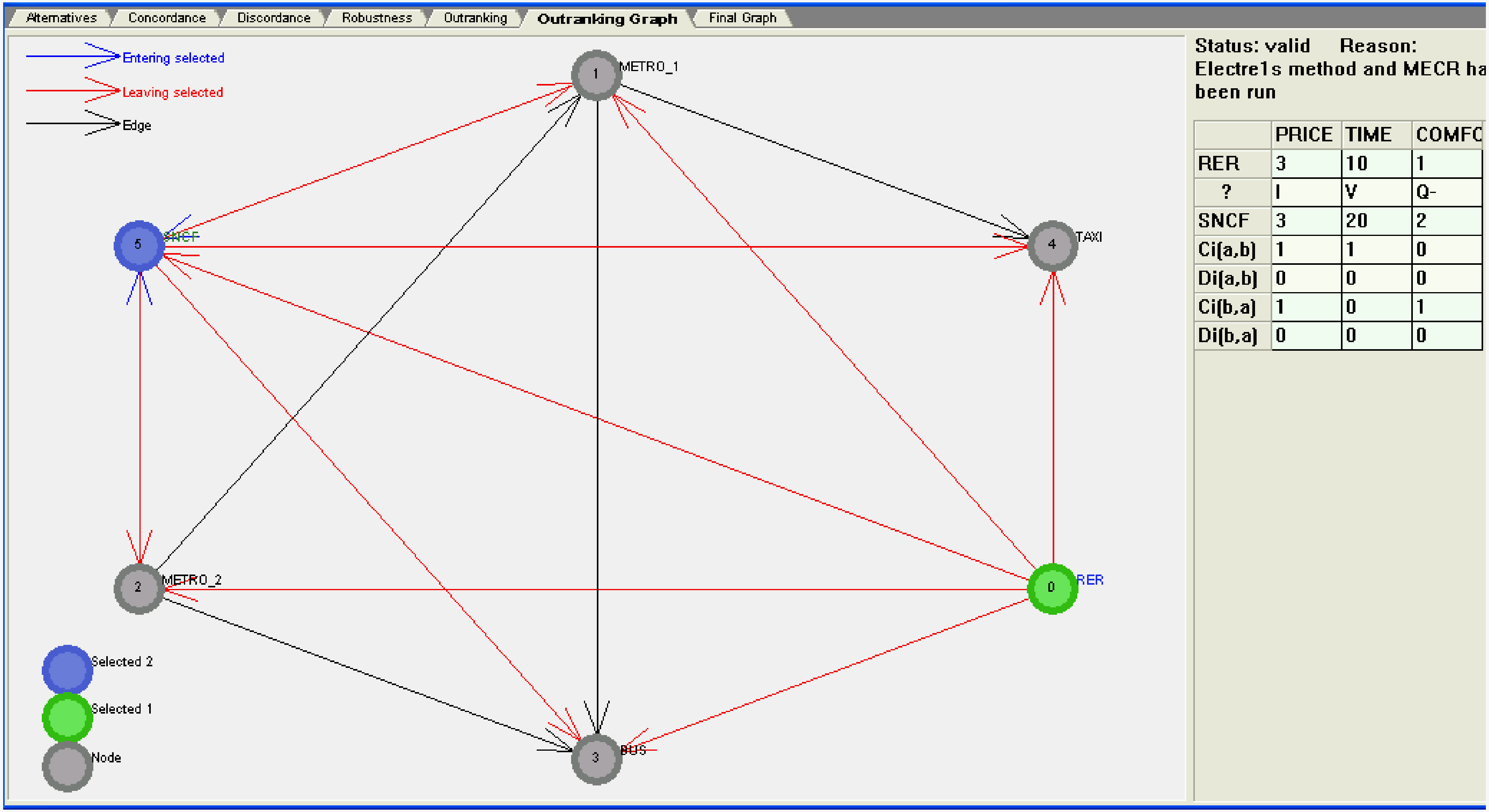
ELECTRE Is - example



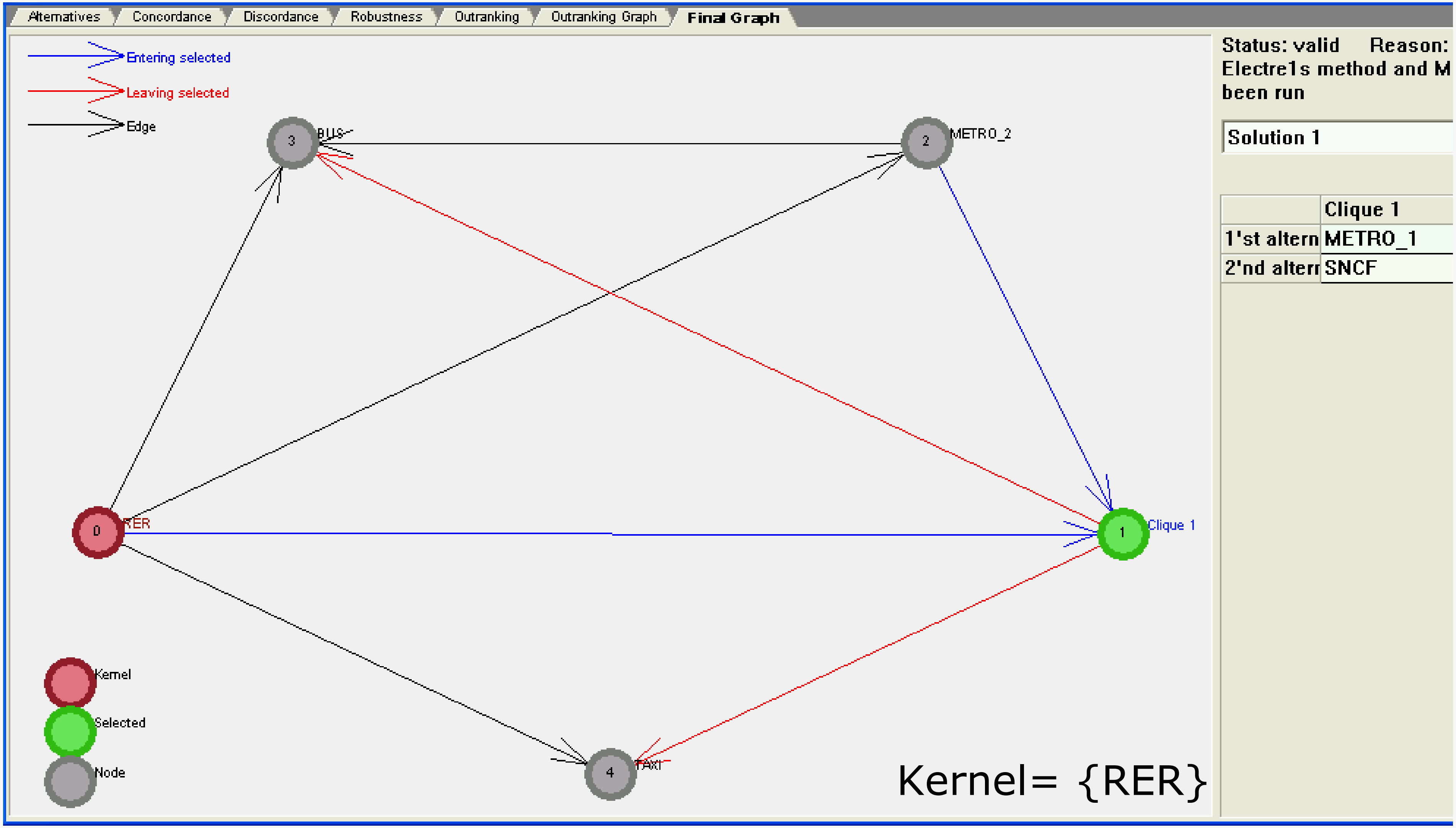
ELECTRE Is - example



ELECTRE Is - example

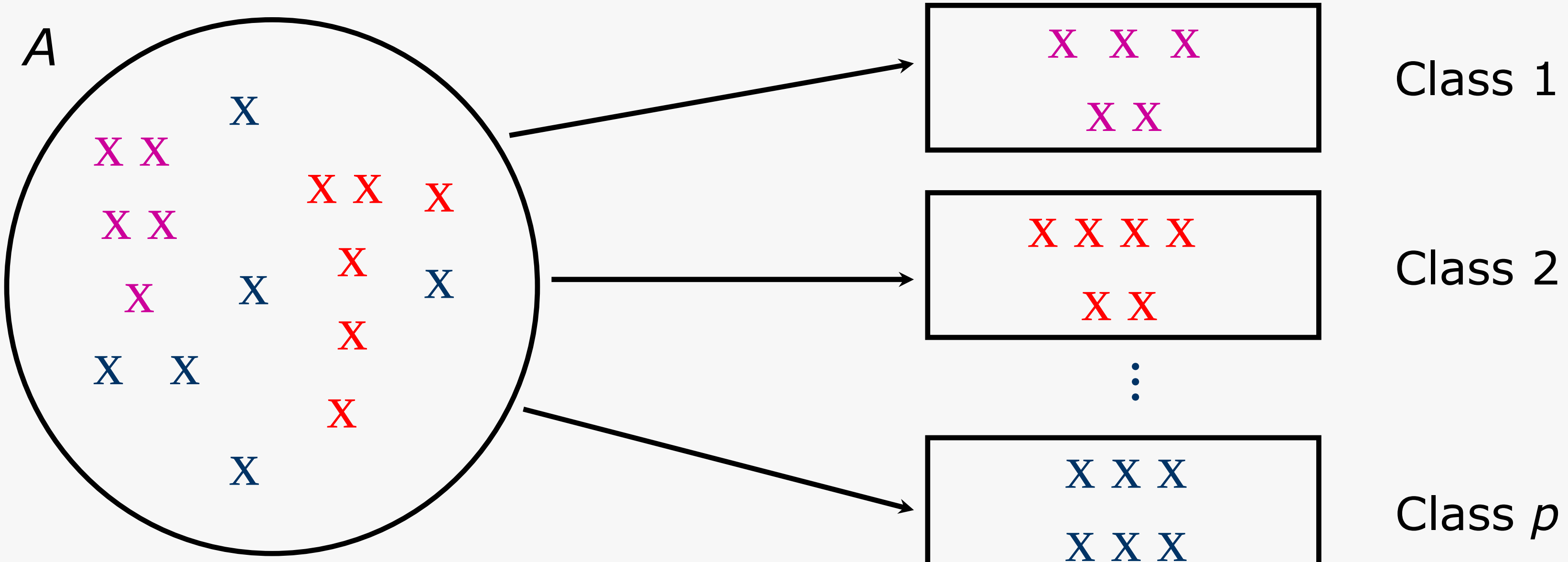


ELECTRE Is - example



Kernel= {RER}

ELECTRE TRI: sorting problem (P_β)

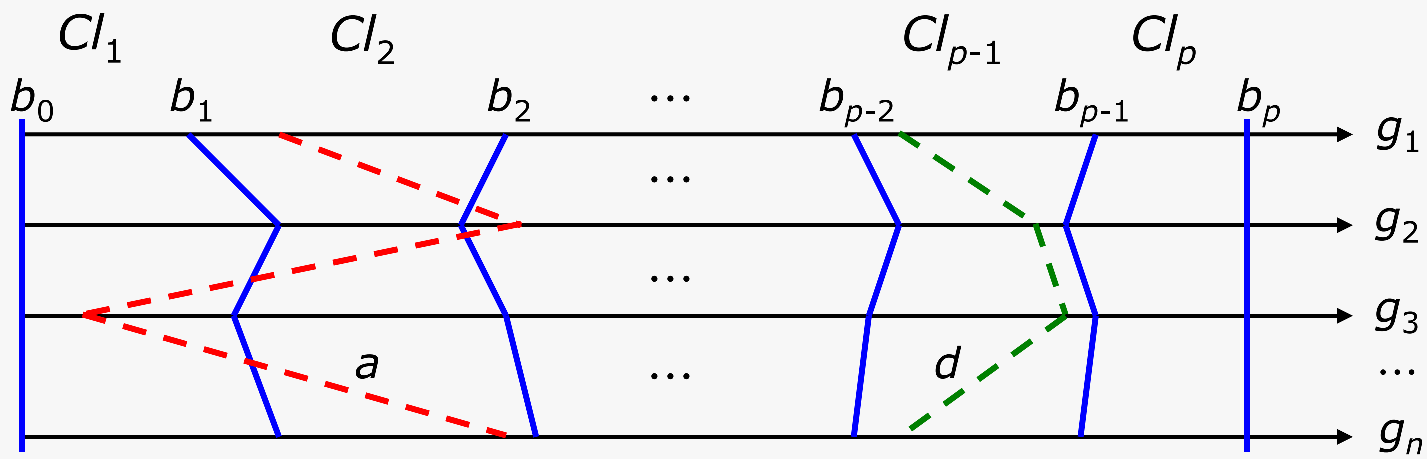


Class 1 $\succ$ Class 2 $\succ$ … $\succ$ Class p

ELECTRE TRI

- Input data: finite set of actions $A=\{a, b, c, \ldots, h\}$
  consistent family of criteria $G=\{g_1, g_2, \ldots, g_n\}$
  preference-ordered decision classes Cl_t, $t=1,\ldots,p$

Decision classes are caracterized by limit profiles b_t, $t=0,1,\ldots,p$



- The preference model, i.e. outranking relation S is constructed for each couple (a, b_t), for every $a \in A$ and $t=0,1,\ldots,p$

ELECTRE TRI

- Preferential information ($i=1,\dots,k$)

  intracriteria: – indifference thresholds q_i^t for each class limit, $t=0,1,\dots,p$

  – preference thresholds p_i^t for each class limit, $t=0,1,\dots,p$

  intercriteria: – importance coefficients (weights) of criteria k_i

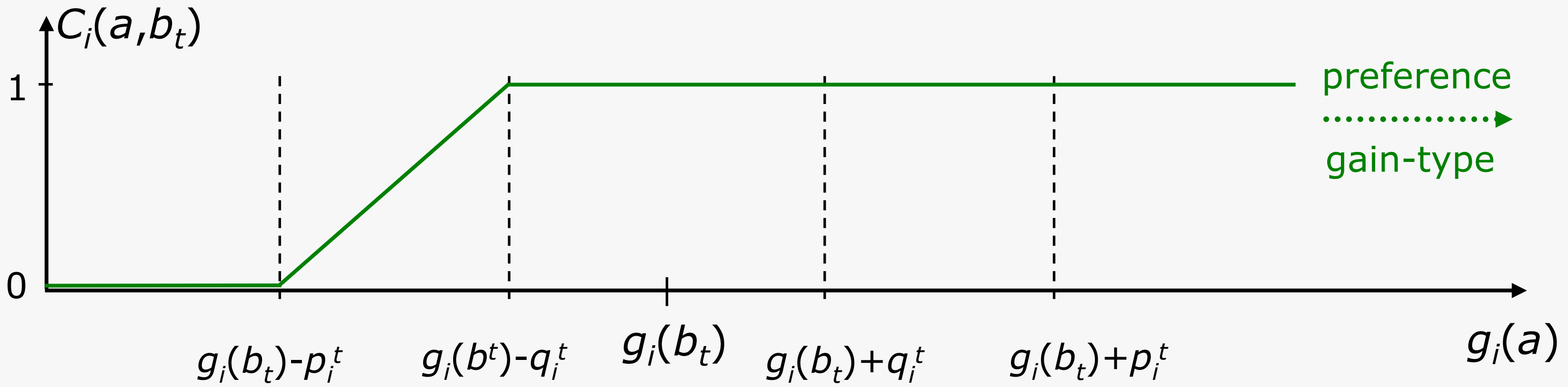
  – veto thresholds for each class limit v_i^t, $t=0,1,\dots,p$

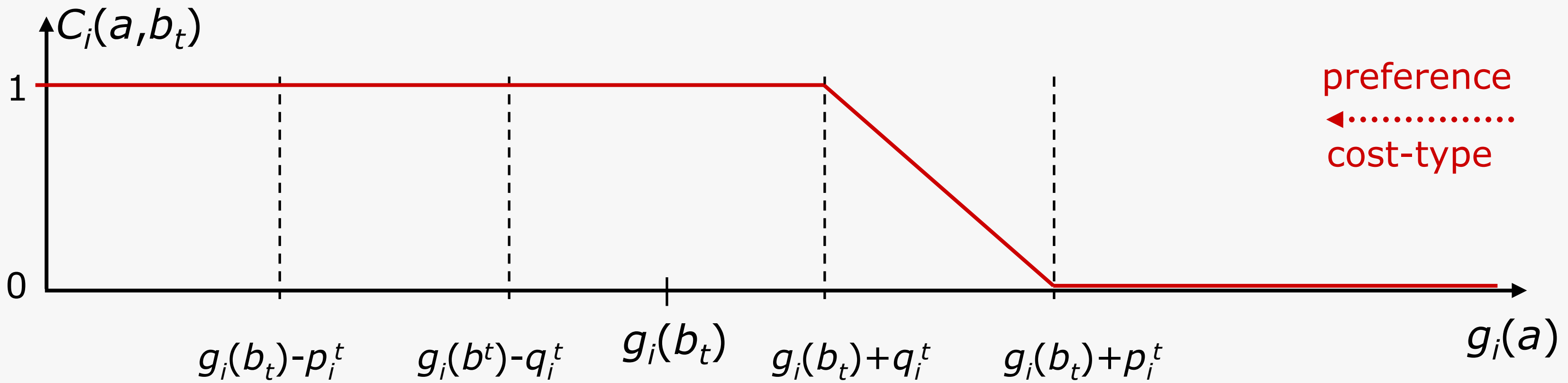
- $0 \le q_i^t \le p_i^t \le v_i^t$ are constant for each class limit b_t, $t=0,1,\dots,p$

- Concordance and discordance tests of ELECTRE TRI validate or invalidate the assertions aSb_t and b_tSa

ELECTRE TRI - concordance test

- Checks how strong is the coalition of criteria concordant with the hypothesis aSb_t (for each couple (a,b_t), $\forall a \in A$ and $t=0,1,\ldots,p$)
- Concordance coefficient $C_i(a,b_t)$ for each criterion g_i:





ELECTRE TRI - concordance test

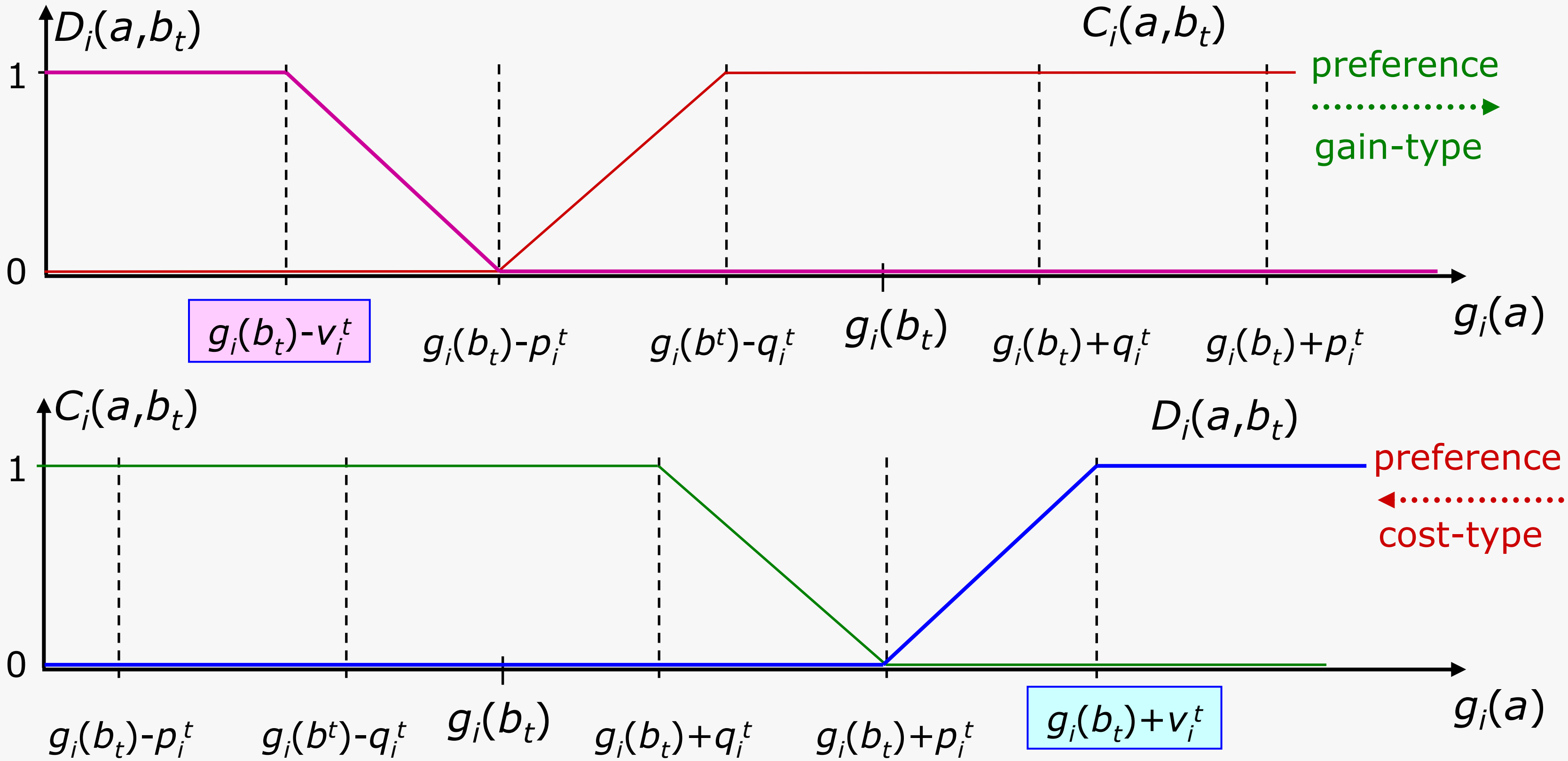
- Aggregation of concordance coefficients for $a,b \in A$:

$$C(a,b_t) = \frac{\sum_{i=1}^{n} k_i C_i(a,b_t)}{\sum_{i=1}^{n} k_i}$$

- $C(a,b_t) \in [0, 1]$

ELECTRE TRI - discordance test

- Checks how strong is the coalition of criteria discordant with the hypothesis aSb_t (for each couple (a,b_t), $\forall a \in A$ and $t=0,1,\ldots,p$)
- Discordance coefficient $D_i(a,b_t)$ for each criterion g_i:



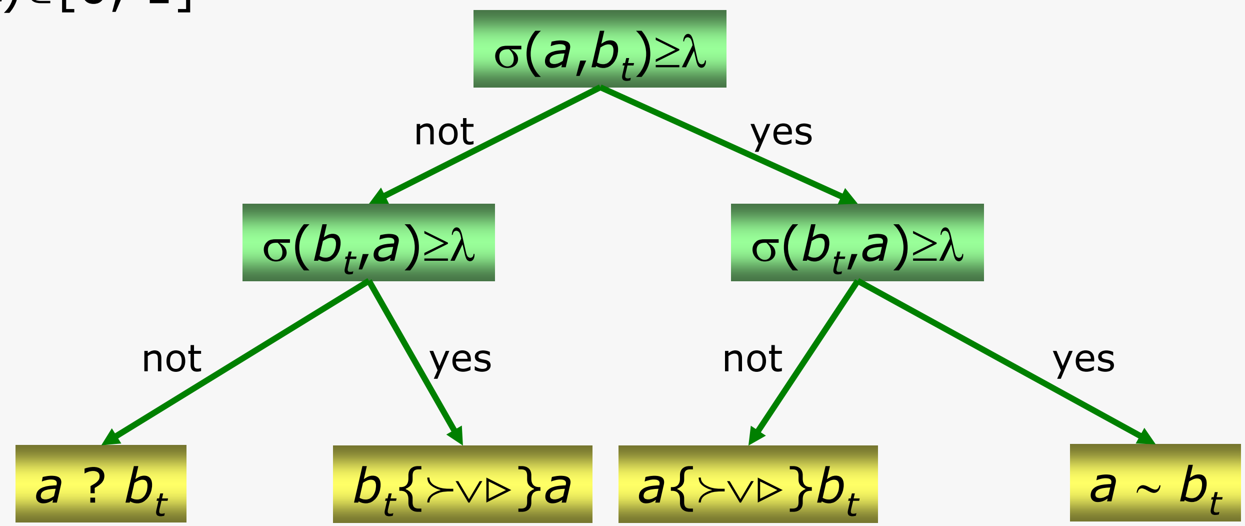
ELECTRE TRI – credibility of the outranking relation

- Conclusion from concordance and discordance tests – credibility of the outranking relation:

$$\sigma(a,b_t) = C(a,b_t)\prod_{i\in F}\frac{1-D_i(a,b_t)}{1-C(a,b_t)}$$

where $F = \{i : D_i(a,b_t) > C(a,b_t)\}$

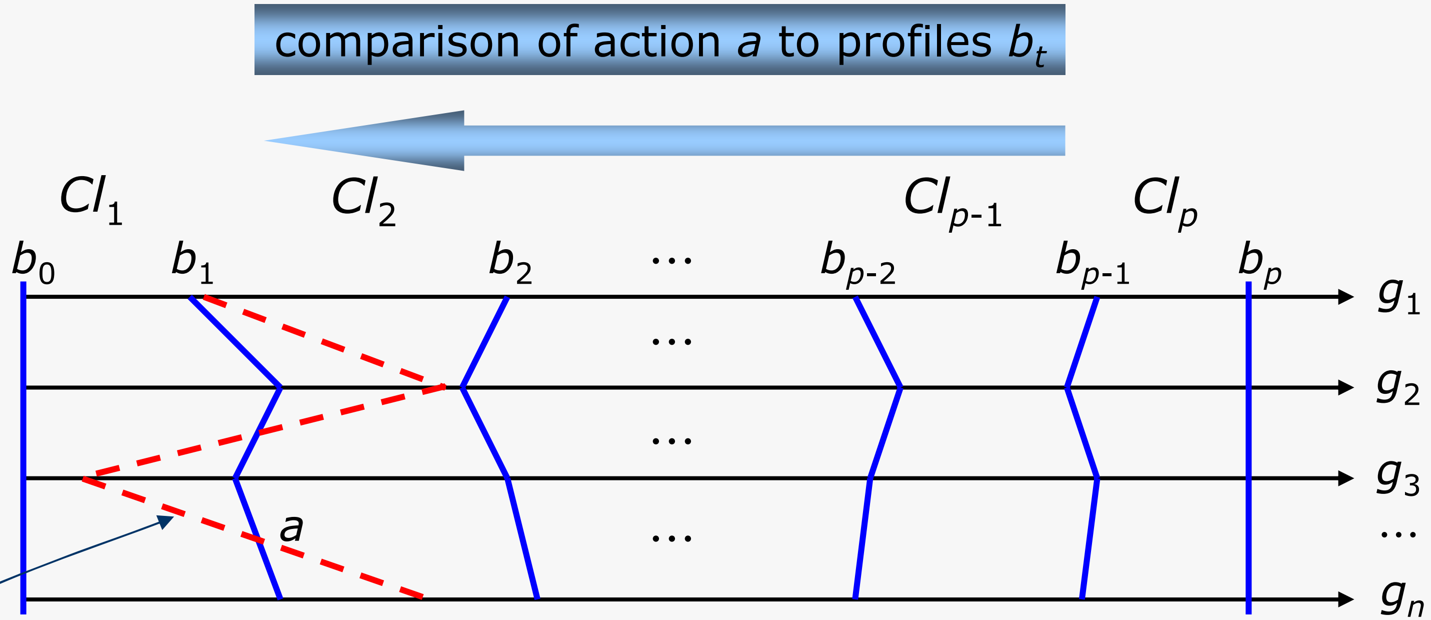
- $\sigma(a,b_t)\in[0,\ 1]$



- What is the assignment of actions to decision classes ?

ELECTRE TRI – exploitation of *S* and final recommendation

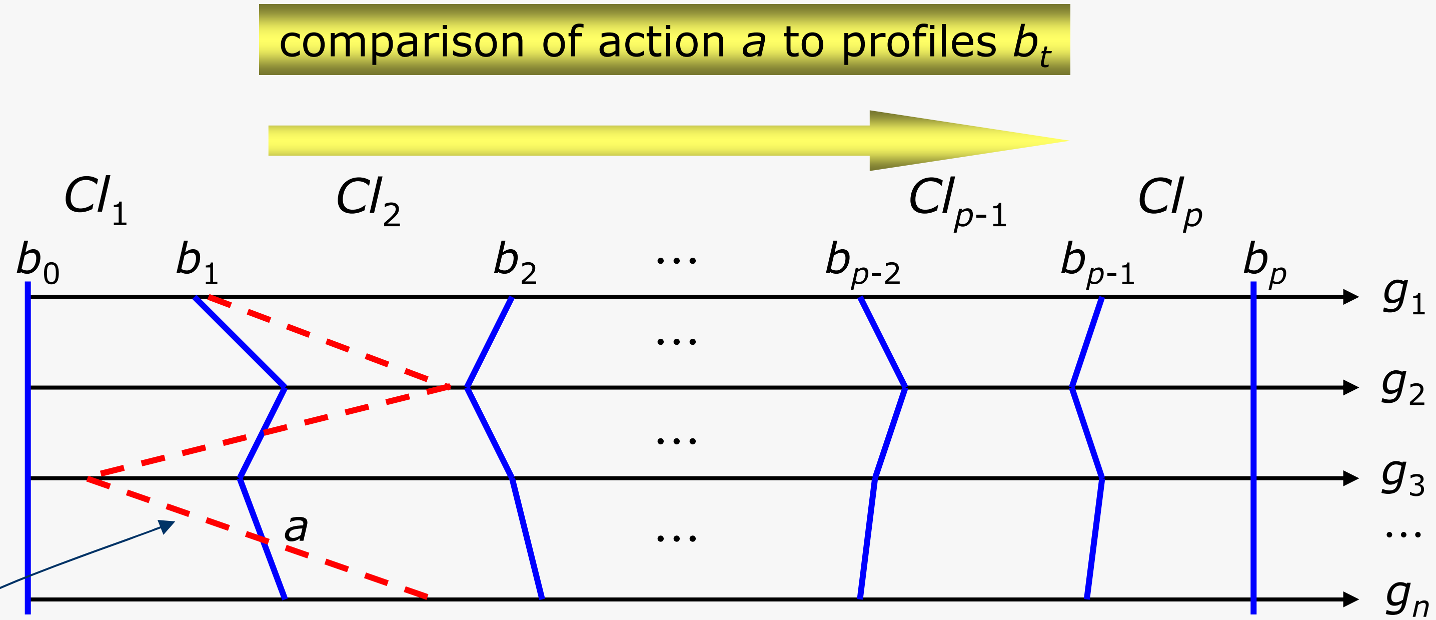
- Assignment of actions to decision classes based on two ways of comparison of actions $a \in A$ to limit profiles b_t, $t=0,1,\ldots,p$
  - ***Pessimistic***: compare action a successively to each profile b_t, $t=p-1,\ldots,1,0$; if b_t is the first profile such that aSb_t, then $a \to Cl_{t+1}$



  - Pessimistic, because for zero thresholds, $a \to Cl_{t+1} \Leftrightarrow g_i(a) \geq g_i(b_t)\ \forall i$, e.g. $a \to Cl_{\mathbf{1}}$

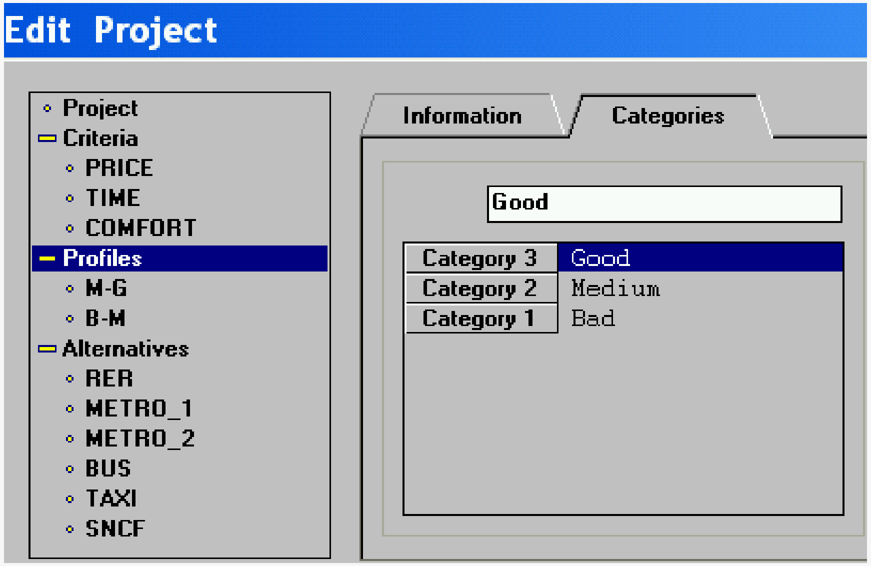
ELECTRE TRI – exploitation of S and final recommendation

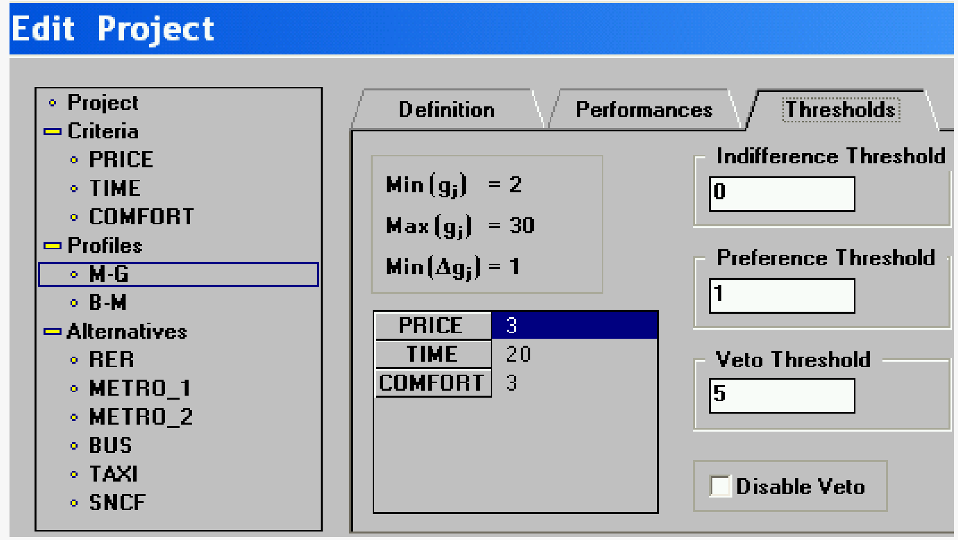
- Assignment of actions to decision classes based on two ways of comparison of actions $a \in A$ to limit profiles b_t, $t=0,1,\ldots,p$
  - ***Optimistic***: compare action a successively to each profile b_t, $t=1,\ldots,p$; if b_t is the first profile such that $b_t\{\succ \vee \triangleright\}a$, then $a \rightarrow Cl_t$



  - Optimistic, because for zero thresholds, $a \rightarrow Cl_t \Leftrightarrow g_i(b_t)<g_i(a)\ \forall i$, e.g. $a \rightarrow Cl_{\mathbf{2}}$

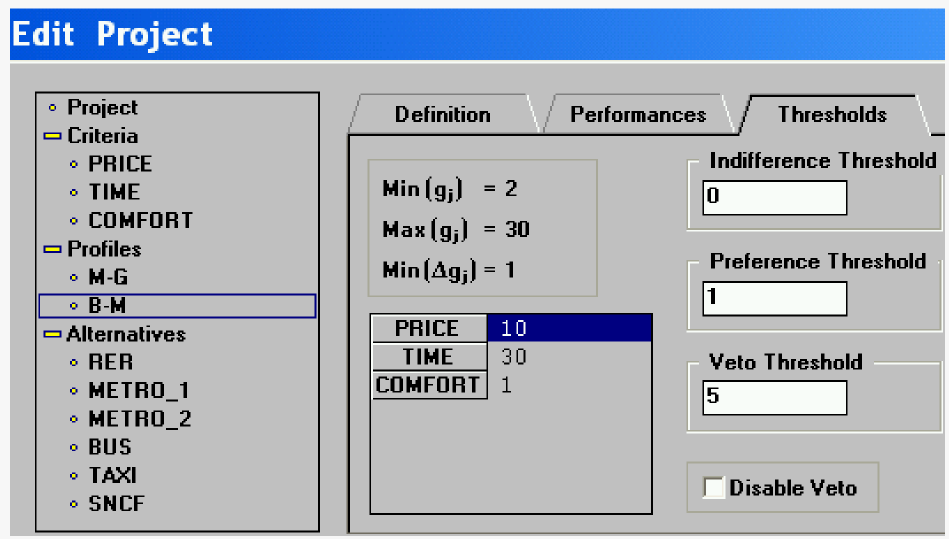
ELECTRE TRI - example





Performances of Alternatives

	PRICE	TIME	COMFORT
RER	3	10	1
METRO_1	4	20	2
METRO_2	2	20	0
BUS	6	40	0
TAXI	30	30	3
SNCF	3	20	2



$k_{PRICE}=3$

$k_{TIME}=5$

$k_{COMFORT}=2$

$\lambda=0.75$

ELECTRE TRI - example

Comparison to Profile

	B-M	M-G
RER	>	>
METRO_1	>	<
METRO_2	>	R
BUS	R	<
TAXI	<	<
SNCF	>	I

Assignment by Category

Category Name	Pessimistic Assignment	Optimistic Assignment
Good	RER	RER
Medium	SNCF	METRO_2
Bad		SNCF

Category Name	Pessimistic Assignment	Optimistic Assignment
Good	METRO_1	METRO_1
Medium	METRO_2	BUS
Bad		

Category Name	Pessimistic Assignment	Optimistic Assignment
Good	BUS	TAXI
Medium	TAXI	
Bad		

Statistics of Assignment

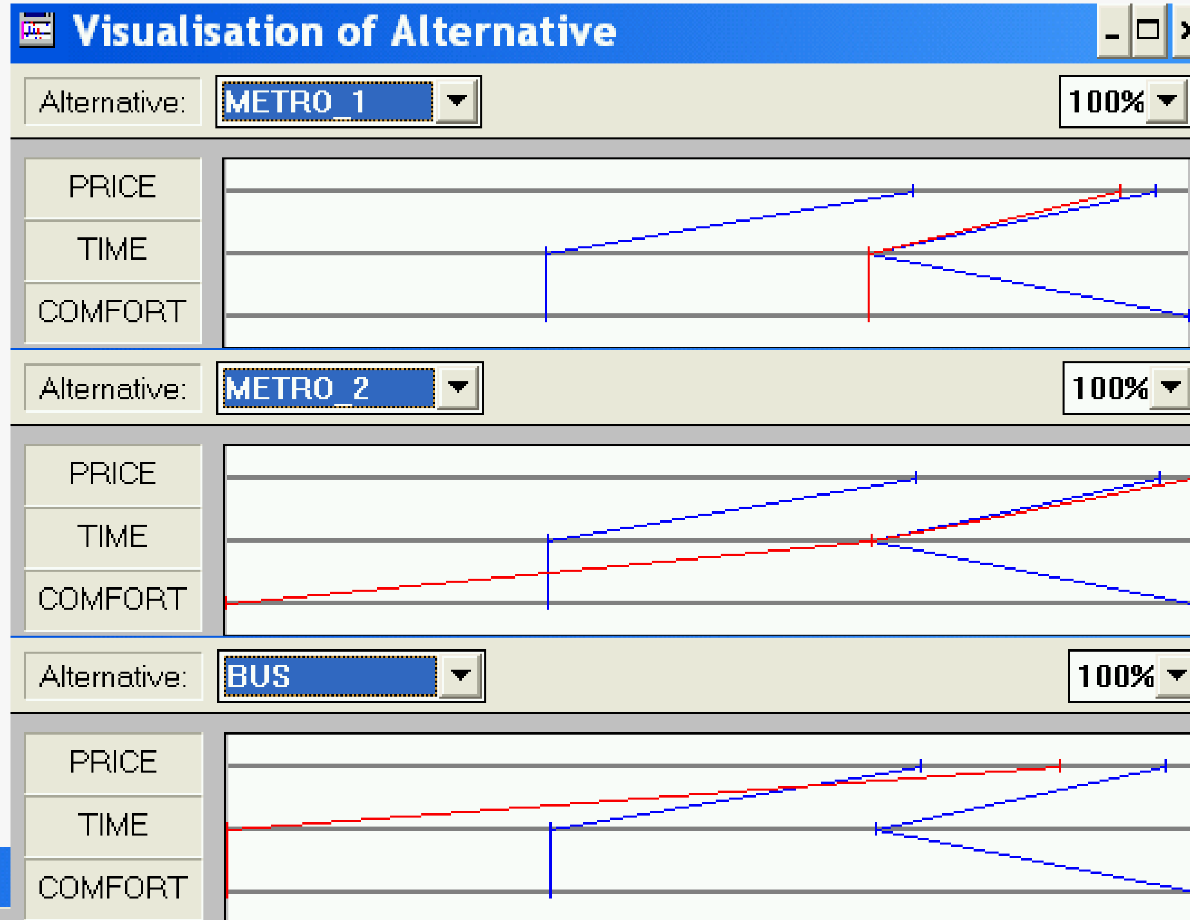
Category Name	Pessimistic Assignment	Optimistic Assignment
Good	33 % (2 of 6)	50 % (3 of 6)
Medium	33 % (2 of 6)	33 % (2 of 6)
Bad	33 % (2 of 6)	17 % (1 of 6)

ELECTRE TRI - example

Degrees of Credibility

	B-M	M-G
RER	1.000	0.800
	0.000	0.500
METRO_1	1.000	0.700
	0.000	1.000
METRO_2	1.000	0.000
	0.000	0.700
BUS	0.500	0.000
	0.583	1.000
TAXI	0.000	0.000
	0.800	1.000
SNCF	1.000	1.000
	0.000	1.000

$\lambda=0.75$



Assignment by Alternative

Alternative Name	Pessimistic Assignment	Optimistic Assignment
RER	Good	Good
METRO_1	Medium	Medium
METRO_2	Medium	Good
BUS	Bad	Medium
TAXI	Bad	Bad
SNCF	Good	Good