On enumerating the kernels in a bipolar-valued outranking digraph

Raymond BISDORFF *

Abstract

In this communication we would like to thoroughly cover the problem of computing all kernels, i.e. minimal outranking and/or outranked independent choices in a bipolar-valued outranking digraph. First we present and discuss several algorithms for enumerating the kernels in a crisp digraph. The second part will be concerned with extending these algorithms in order to compute all valued kernels in the corresponding bipolar-valued outranking digraph.

Key words : Graph Theory, Maximum Independent Sets, Enumerating Kernels, Outranking Digraphs

Introduction

Minimal independent and outranking or outranked choices, i.e. kernels, in valued outranking digraphs are an essential formal tool for solving best unique choice problems in the context of our multicriteria decision aid methodology [9]. It appears, following recent formal results [8], that computing these kernels may rely on the enumeration of all maximal independent sets in the associated crisp median cut outranking digraph. In this article we shall therefore first present the bipolar-valued concepts of outranking digraphs, minimal independent outranking and outranked choices, each associated with their corresponding median cut crisp concept. In a second section, we shall then present known and new algorithms for directly enumerating these crisp choices in a bipolar-valued digraph. A third section will be devoted to extending these algorithms in order to compute the corresponding bipolar-valued choices.

* Applied Mathematics Unit, University of Luxembourg, 162a, avenue de la Faïencerie, L-1511 Luxemborg, `http://sma.uni.lu/bisdorff`

1 Kernels in bipolar valued directed graphs

1.1 Bipolar valued outranking graphs

We consider a binary outranking relation S defined on a set $X : \{x, y, z, \ldots\}$ of decision actions. S represents relational statements supporting a "*to be at least as good*" preference situation we may observe between the decision actions given in X ([17]).

S is characterised in an ordinal bipolar credibility or robustness domain: $\tilde{S} : X \times X \rightarrow \mathcal{L} : \{-m, \ldots, 0, \ldots, m\}$, with m integer, ≥ 1 and finite. $\tilde{S}(x, y) > 0$ signifies that the statement $x \, \mathrm{S} \, y$, i.e. *"decision action x is at least as good as decision y"* is *more true than false*. $\tilde{S}(x, y) < 0$ signifies that the same statement $x \, \mathrm{S} \, y$ is *more false than true*. $\tilde{S}(x, y) = 0$ signifies that the truth denotation of statement $x \, \mathrm{S} \, y$ is *suspended*, i.e. statement $x \, \mathrm{S} \, y$ is *logically undetermined*. For short we say $x \, \mathrm{S} \, y$ is either $\mathcal{L}$-true, $\mathcal{L}$-false or $\mathcal{L}$-undetermined.

We shall distinguish the smallest possible $\mathcal{L}$ domain showing three values: $\{-1, 0, 1\}$, denoted $\mathcal{L}_3$. Similarly, we shall distinguish in particular, the degenerated bi-valued domain $\mathcal{B} = \{-1, 1\}$, missing the undetermined denotation, which corresponds to a classic "truth"- and "falsity"-valued Boolean characteristic domain. Similarly we shall denote $\mathcal{L}_{/0}$ a bipolar-valued characteristic domina missing the undetermined value 0.

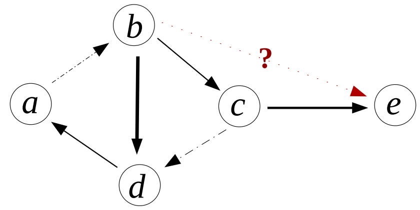
We denote $G^{\mathcal{L}}(X, \tilde{S})$ the digraph representing an $\mathcal{L}$-valued outranking relation, where $\tilde{S} : X \times X \rightarrow \mathcal{L}$. The semiotic richness of the $\mathcal{L}$-valued characterisation of S is not easily representable in diagrams where a binary relation either exists or not.

To recover such a logical determination we use the following logical polarizations of $\mathcal{L}$-valued digraphs. Let $G^{\mathcal{L}}(X, \tilde{S})$ be a $\mathcal{L}$-valued We denote $G(X, S)$ its associated *strict median cut* crisp digraph where $\forall x, y \in X : \tilde{S}(x, y) > 0 \Rightarrow (x, y) \in S$, $\tilde{S}(x, y) < 0 \Rightarrow (x, y) \notin S$.

Furthermore, we shall call *$\mathcal{L}$-determined* a digraph $G^{\mathcal{L}}(X, \tilde{S})$ such that $\tilde{S}(x, y) \neq 0$ for all $x, y \in X$ and $x \neq y$.

Example 1 (B. Roy (2005), private communication)**.**
Let $G_1^{\mathcal{L}}(X, \tilde{S}_1)$ be the bipolar valued digraph where: $X_1 = \{a, b, c, d, e\}$,
$\mathcal{L} = \{-10, \ldots, 0, \ldots, 10\}$ and $\tilde{S}_1$ is given as follows:

$\tilde{S}_1$	a	b	c	d	e
a	-	**6**	-10	-7	-9
b	-8	-	**9**	**10**	**0**
c	-10	-10	-	**6**	**9**
d	**8**	-8	-10	-	-7
e	-10	-9	-7	-8	-



The associated strict median cut digraph

The above strict median cut technique polarizes to *true* all outranking statements that are $\mathcal{L}$-true, and to *false* all outranking statements that are $\mathcal{L}$-false. In the example (1), we may notice that $\tilde{S}_1(b, e) = 0$, i.e. outranking statement bSe may indeed be true *or* false. And the associated digraph $G(X, S)$ is a partially defined digraph. It is worthwhile noticing that in case of an $\mathcal{L}$-determined digraph, the strict median cut polarization gives a classic binary relation S on X.

In the sequel, we only consider finite sets X of decision actions so that all the digraphs we consider are finite. The cardinality $n = |X|$ of X gives the *order*. The cardinality $s = |S|$ – the number of $\mathcal{L}$-true arcs of the graph – gives the *size* of $G^{\mathcal{L}}$. All outranking digraphs we consider are naturally reflexive, so that we generally ignore the reflexive terms, except if explicitly mentioned. $s/n(n-1) \times 100$ gives the *fill rate* (in %) of $G^{\mathcal{L}}$. We call $G^{\mathcal{L}}$ a connected digraph if the symmetric and transitive closure of $G(X, S)$ corresponds to a complete graph. In fact a connected graph is a graph that contains no isolated vertices. In example (1) the digraph $G_1^{\mathcal{L}}(X_1, \tilde{S}_1)$ is connected, of order 5, of size 5, of fill rate $6/20 \times 100 = 30\%$.

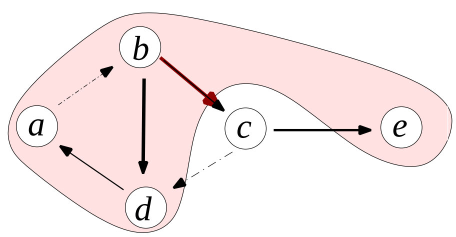
1.2 Outranking and outranked choices

Let $G^{\mathcal{L}}(X, \tilde{S})$ be a bipolar valued outranking graph.

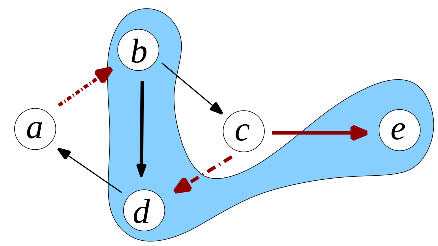
A *choice* Y in $G^{\mathcal{L}}$ is a non empty subset Y of X. The set of all possible choices in $G^{\mathcal{L}}$ is the powerset of X, denoted $\mathcal{P}(X)$, except the empty set. We call *single* a minimal choice reduced to a singleton and *greedy* the maximal possible choice, i.e. the whole set X.

Definition 1 (Outranking and outranked choices)**.**
A *choice* Y in $G^{\mathcal{L}}$ is an *outranking* choice if and only if $\forall x \in X : x \notin Y \Rightarrow \exists y \in Y : \tilde{S}(y, x) > 0$. Similarly, A *choice* Y in $G^{\mathcal{L}}$ is an *outranked* choice if and only if $\forall x \in X : x \notin Y \Rightarrow \exists y \in Y : \tilde{S}(x, y) > 0$.

Example 2 (Choices in $G_1^{\mathcal{L}}$)**.**



$\{a, b, d, e\}$ is a outranking choice.

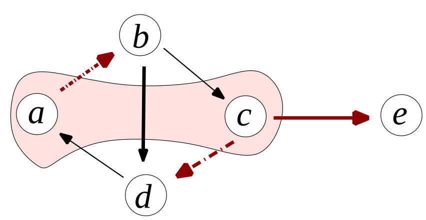


$\{b, d, e\}$ is an outranked choice.

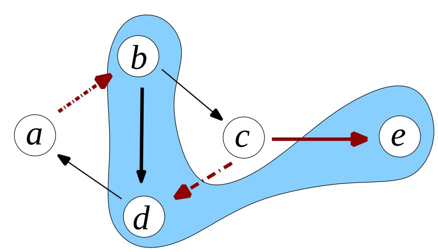
In example (2), we may notice that the choice $\{a, b, d, e\}$ in $G_1^{\mathcal{L}}$ (see example 1) may be reduced without loosing the property of being outranking. The outranked choice in the same example (2) may not however be reduced without loosing its outrankedness property. Minimal or maximal cardinality of choices with respect to a given qualification is formally captured in the following definition.

Definition 2 (Minimal and maximal choices)**.**
A choice Y in $G^{\mathcal{L}}$, verifying a property P, is *minimal* with this property whenever, $\forall Y' \in G^{\mathcal{L}}$ which verify the same property P, we have $Y' \not\subseteq Y$. Similarly, a choice Y in $G^{\mathcal{L}}$, verifying a property P, is *maximal* with this property whenever, $\forall Y' \in G^{\mathcal{L}}$ which verify property P, we have $Y' \not\supseteq Y$.

Example 3 (Minimal choices in $G_1^{\mathcal{L}}$)**.**



$\{a, c\}$ is a minimal outranking choice.



$\{b, d, e\}$ is a minimal outranked choice

Comparing the outranking choice $\{a, b, d, e\}$ in example (2) with the minimal outranking choice $\{a, c\}$ in example (3) we may notice that minimality of outrank is related to the neighbourhoods of the nodes of the digraph.

Definition 3 (Open and closed neighbourhoods)**.**
We denote $N^+(x) = \{y \in X \;/\; \tilde{S}(x, y) > 0\}$ the *open outranked neighbourhood* of a node $x \in X$. We denote $N^+[x] = N^+(x) \cup \{x\}$ the *closed* outranked neighbourhood of x. We denote $N^-(x) = \{y \in X \;/\; \tilde{S}(y, x) > 0\}$ the open *outranking* neighbourhood of a node x. We denote $N^-[x] = N^-(x) \cup \{x\}$ the closed outranking neighbourhood of x.

The neighbourhood concept may easily be extended to a choice.

Definition 4 (Choice neighbourhoods)**.**
The closed and open *outranked neighbourhood* of a choice Y in $G^{\mathcal{L}}$ are given by the union of the respective neighbourhoods of the members of the choice:

$$N^+[Y] = \bigcup_{x \in Y} N^+[x], \quad N^+(Y) = \bigcup_{x \in Y} N^+(x). \tag{1}$$

The closed and open *outranking neighbourhood* of a choice Y in $G^{\mathcal{L}}$ are similarly given by the union of the respective elementary outranking neighbourhoods:

$$N^-[Y] = \bigcup_{x \in Y} N^-[x], \quad N^-(Y) = \bigcup_{x \in Y} N^-(x). \tag{2}$$

Definition 5 (Private neighbourhood)**.**
The (closed) *private outranked neighbourhood* $N^+_Y[x]$ of a node x in a choice Y containing x is defined as follows: $N^+_Y[x] = N^+[x] - N^+[Y - \{x\}]$. Similarly, the (closed) private *outranking* neighbourhood $N^-_Y[x]$ of a node x in a choice Y is defined as follows: $N^-_Y[x] = N^-[x] - N^-[Y - \{x\}]$. In case of a single choice, both the outranked and the outranking neighbourhood are considered to be private by convention.

In the outranking choice $Y = \{a, b, d, e\}$ of example (2), we may notice that action a for instance has no private outranked neighbourhood. Indeed $N^+_Y[a] = N^+[a] - N^+[Y - \{a\}]$ where $N^+[a] = \{a, b\}$ and $N^+[Y - \{a\}] = X$. Action b however has action c as private outranked neighbourhood. The concept of private neighbourhoods leads us naturally to the notion of irredundant choices.

Definition 6 (Irredundant choice)**.**
A outranking choice Y in $G^{\mathcal{L}}$ is called +*irredundant* if and only if all its members have a non empty private outranked neighbourhood, i.e. $\forall x \in Y : N^+_Y[x] \neq \emptyset$. Similarly, an outranked choice Y in $G^{\mathcal{L}}$ is called -*irredundant* if and only if all its members have a private outranking neighbourhood, i.e. $\forall x \in Y : N^-_Y[x] \neq \emptyset$.

In example (3), the outranking choice $\{a, c\}$ is +irredundant as $N^+_{\{a,c\}}[a] = \{a, b\}$ and $N^+_{\{a,c\}}[c] = \{c, d, e\}$. Similarly the outranked choice $\{b, d, e\}$ is -irredundant as $N^-_{\{b,d,e\}}[b] = \{a, b\}$, $N^-_{\{b,d,e\}}[d] = \{c\}$ and $N^-_{\{b,d,e\}}[e] = \{e\}$.

Minimality of outrankingness (resp. outrankedness) and maximality of +irredundancy (resp. -irredundancy) are evidently linked.

Proposition 1.
(i) A outranking (resp. outranked) choice Y in $G^{\mathcal{L}}$ is minimal outranking (resp. outranked) if and only if it is outranking (resp. outranked) and +irredundant (resp. -irredundant) (Cockayne, Hedetniemi, Miller 1978).

(ii) Every minimal outranking (resp. outranked) choice Y in $G^{\mathcal{L}}$ is maximal +irredundant (resp. -irredundant) (Bollobás, Cockayne, 1979).

Proof. Property (i) following easily from property (2), we demonstrate only the latter one.

[$\Rightarrow$] Let us suppose that Y is minimal outranking but not maximal +irredundant. This implies that there exists a node $x \in X - Y$ such that $Y \cup \{x\}$ is +irredundant, i.e. $N^+(Y)$ is a proper subset of $N^+(Y \cup \{x\})$. This contradicts however the fact that Y is outranking.

[$\Leftarrow$] The other way round, let us suppose that Y is maximal +irredundant but not minimal outranking. This implies that there must exist an $y \in Y$ such that $Y - \{y\}$ still remains outranking, i.e. this y cannot have a private outranked neighbourhood with respect to Y. This contradicts however the hypothesis that Y is +irredundant.

A similar reasoning is valid for outranked and -irredundant choices. □

1.3 Dominant and outranked kernels in a digraph

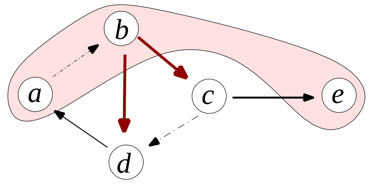
Definition 7 (Independent choices)**.**
A choice Y in $G^{\mathcal{L}}$ is called *independent* if and only if for all $x, y \in Y : \tilde{S}(x, y) < 0$. A outranking (resp. outranked) and independent choice Y in $G^{\mathcal{L}}$ is called an *outranking (resp. outranked) kernel* of the graph $G^{\mathcal{L}}$.

Independence and the outranking or outranked property are tightly related.

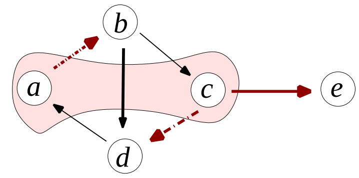
Proposition 2 (Berge, 1958)**.**
Let $G^{\mathcal{L}}(X, \tilde{S})$ be an $\mathcal{L}$-determined digraph. (i) Every kernel is a minimal outranking (resp. outranked) choice. (ii) Every minimal outranking (resp. outranked) and independent choice is maximal independent.

Proof. (1) Let us suppose that a outranking kernel Y is indeed not a minimal outranking (respectively outranked) choice. This implies that there exists a outranking (respectively outranked) choice $Y' \subset Y$ such that Y' is still outranking (respectively outranked). This implies that $\forall y \in Y - Y'$ there must exist some $y' \in Y'$ such that $(y, y') \in S$ (respectively $(y', y) \in S$. This is contradictory with the fact that Y is independent. (2) Let us suppose that a outranking kernel Y is indeed not a maximal independent choice. This implies that there must exist a $Y' \supset Y$ such that Y' is still independent. But Y is by hypothesis a outranking (respectively outranked) choice, i.e. $\forall y' \in Y' - Y$ there must exist some $y \in Y$ such that $(y, y') \in S$ (respectively $(y', y) \in S$. Hence there appears again a contradiction. □

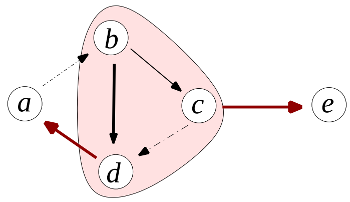
Not all minimal outranking (resp. outranked) choices are independent, i.e. kernels. In digraph $G_1^{\mathcal{L}}$ of example 1, for instance, we observe the following four minimal outranking choices, of which only choice $\{a, c\}$ is independent and therefore a outranking kernel.



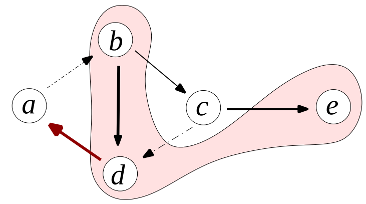
minimal outranking choice,



outranking kernel,



minimal outranking choice.



minimal outranking choice.

Kernels and minimal choices however coincide in $\mathcal{L}$-determined and transitive digraphs.

Proposition 3.
Let $G^{\mathcal{L}}(X, \tilde{S})$ be a $\mathcal{L}$-transitive and $\mathcal{L}$-determined digraph, i.e. the associated crisp graph $G(X, S)$ supports a transitive outranking relation S. A choice Y in $G^{\mathcal{L}}$ is a outranking (resp. outranked) kernel if and only if Y verifies one of the following equivalent conditions:
(i) Y is minimal outranking (resp. outranked);
(ii) Y is outranking (resp. outranked) and independent; (iii) Y is outranking (resp. outranked) and +(-)irredundant. (iv) Y is maximal +(-)irredundant.

Proof. (i) $\Leftrightarrow$ (iii) $\Leftrightarrow$ (iv) are covered by proposition 1, and (ii) $\Rightarrow$ (i) is covered by proposition 2. We only need to prove that (i) $\Rightarrow$ (ii).

Let us therefore suppose that a minimal outranking (respectively outranked) choice Y is indeed not independent. As $G^{\mathcal{L}}$ is $\mathcal{L}$-determined, this implies that there exists some proper subset $Y' \subset Y$ such that for $y \in Y - Y'$ and $y' \in Y'$ we observe $(y, y') \in S$ (respectively $(y', y) \in S$. As Y is a minimal outranking (respectively outranked) choice, each action in Y must have a private outranked (resp. outranking) neighbourhood and in particular all actions in Y'. By transitivity of S, the private neighbourhoods $N_Y^+(y')$ and $N_Y^-(y')$ of an action $y' \in Y'$ are transferred to $y \in Y - Y'$. And $Y - Y'$ remains therefore a outranking (resp. outranked) choice. This is however contradictory with the hypothesis that Y is minimal with this quality. □

It is worthwhile noticing that proposition (3) only applies to $\mathcal{L}$-determined digraphs. In case we observe a partially determined graph, it may happen that a minimal outranking (resp. absobent) choice is not effectively independent, and vice-versa, it may indeed happen that a maximal independent choice is neither outranking nor outranked. All depends upon the particular presence of $\mathcal{L}$-undetermined relations.

We have not the space in this paper to present all existence results for kernels in a digraph (see for instance [13]). Relevant properties for our purpose are summarized below, where we generally suppose that the graph is characterized in $\mathcal{L}_{/0}$.

1. Every digraph supports minimal outranking (resp. outranked) choices.

2. A transitive digraph always supports a outranking (resp. outranked) kernel and all its kernels are of same cardinality (König, 1950 [15]).

3. A symmetric digraph always supports a conjointly outranking and outranked kernel (Berge, 1958 [1]).

4. An acyclic digraph always supports a unique outranking (resp. outranked) kernel (Von Neumann, 1944 [21]).

5. If a digraph does not contain any cordless circuit of odd length, it supports an outranking (resp. outranked) kernel (Richardson, 1953).

2 Enumerating outranking and outranked kernels

2.1 Minimal outranking and outranked choices

Definition 8 (Hereditary properties)**.**
A property P of choices is said to be *hereditary* if whenever a choice Y has property P, so does every proper subchoice $Y' \subset Y$. A property P of choices is said to be *superhereditary* if whenever a choice Y has property P, so does every proper superchoice $Y' \supset Y$.

Proposition 4. *Being outranking or outranked are superhereditary properties of choices in $G^{\mathcal{L}}$. Similarly, Independence, +irredundancy, and -irredundancy are hereditary properties of choices in $G^{\mathcal{L}}$.*

Proof. Hereditary follows immediately from the definition of an independent, an +irredundant, and an -irredundant choice. Superhereditary follows again readily from the definition being outranking or outranked. □

Inheritance of being outranking makes it possible to implement the search for minimal outranking choices as a path algorithm in the outranking choices graph associated with $G^{\mathcal{L}}$.

Definition 9 (P-Choice graphs)**.**
Let $G^{\mathcal{L}}(X, \tilde{S})$ be an outranking graph. Let $\mathcal{P}(X)$ represent the powerset of choices in $G^{\mathcal{L}}$ with property P. The couple $H(\mathcal{P}(X), P)$ is called the P-choice graph associated with $G^{\mathcal{L}}$. Two choices are linked in $H(\mathcal{P}(X), P)$ if they have some common action.

Proposition 5. *The outranking, outranked, +irredundant, -irredundant and independent choice graphs associated with $G^{\mathcal{L}}$ are each strongly connected.*

Proof. As being outranking or outranked are hereditary properties, there necessarily exists a path from every possible minimal outranking (resp. outranked) choice to X, the largest outranking (resp. outranked choice) and vice versa.

Both irredundancies properties being superhereditary, there necessarily exists a path in the corresponding choice graphs from a maximal irredundant choice to each of its single choice members and vice versa. □

Following proposition 5, enumerating all minimal outranking or outranked choices may be implemented as a graph traversal algorithm in the corresponding choic graphs, where we try to explore all paths from the largest outranking (resp. outranked) choice X to the first subchoices which are irredundant.

Algorithm 1 (Enumerating outranking choices)**.**

global $Hist$
$Hist \leftarrow \emptyset$ # initialise the history
$Y_0 \leftarrow X$ # start with the greedy choice
$K_0^+ \leftarrow \emptyset$ # initialise the result
$K^+ \leftarrow$ **MinimalOutrankingChoices** (Y_0, K_0^+)

def MinimalOutrankingChoices (**In:** Y_i outranking, K_i^+; **Out:** K_{i+1}^+)
    $K^+ \leftarrow \emptyset$
    $IRRED \leftarrow$ **True**
    for $[x \in Y_i : N^+_{Y_i}[x] = \emptyset]$: # Retract in turn all redundant nodes
        $IRRED \leftarrow$ **False**
        $Y_{i+1} \leftarrow Y_i - \{x\}$ # Y_{i+1} remains outranking !
        if $Y_{i+1} \notin Hist$:
            $K^+ \leftarrow K^+ \cup$ **MinimalOutrankingChoices** (Y_{i+1}, K^+):
            $Hist \leftarrow Hist \cup \{Y_{i+1}\}$
    if $IRRED$:
        $K_{i+1}^+ + \leftarrow K_i^+ \cup Y$ # Y is +irredundant (and outranking)
    else:
        $K_{i+1}^+ + \leftarrow K_i^+ \cup K^+$
    return K_{i+1}^+

Proof. The algorithm starts with the greedy choice $Y = X$ which is always outranking and an empty set of minimal outranking choices. The procedure **MinimalOutranking-Choices** collects all minimal outranking choices that may be reached from the initial outranking choice Y.

The call invariants of procedure **MinimalOutrankingChoices** are that the choice Y_i is outranking and K_i^+ is a set of minimal outranking choices collected so far.

If Y_i is outranking, then $Y_{i+1} = Y_i - \{x\}$ is constructed only if $N^+_{Y_i}[x] = \emptyset$, i.e. in case x is a +redundant action and Y_{i+1} remains outranking. If no more +redundant actions may be found, the procedure stops the walk. As $Y_0 = X$ is outranking, the algorithm walks therefore only on paths of the outranking choice graph.

Let us suppose that at call i, K_i^+ contains only minimal outranking choices. Two situations may happen. Either the current choice Y_i is irredundant or all redundant actions have been removed in turn. In the first case, we are in the presence of a maximal irredundant and dominating choice, i.e. a minimal outranking choice which is added to to the current set K_i^+. In the second case, all minimal outranking choices when reducing the current choice are first added up in a local result K^+ to be at the end added up to K_i^+. This way, K_{i+1}^+ can only contain minimal outranking choices. As we start with an empty initial collection K_0^+, it is verified that in the end K^+, if not empty, may only contain minimal outranking choices.

Finally, that we algorithm collects all existing minimal outranking choices in $G^{\mathcal{L}}$ follows from the fact that the outranking choice graph is strongly connected and that therefore, starting from the greedy choice X, the algorithm walks necessarily through all outranking choices in $G^{\mathcal{L}}$. The global history we use keeps track of the visited outranking choices and avoids to explore several times the same outranking choice. □

The same algorithm delivers the minimal outranked choices when replacing in the loop the private outranked neighbourhood with the corresponding private outranking neighbourhood. This way, we only walk on outranked choices and collect all minimal outranked choices instead.

Based again on proposition (5), we may design a similar graph traversal algorithm in the irredundant choice graph. This time, we try to explore all paths from the smallest +irredundant (resp. -irredundent) choices – the single choices – to all outranking or outranked choices we may find on our way.

Algorithm 2 (Enumerating maximal irredundant outranking choices)**.**

global $Hist$
$Hist \leftarrow \emptyset$ # initialise the history
$K^+ \leftarrow \emptyset$ # initialise the result

for $x \in X$:
  $Y_0 \leftarrow \{x\}$ # each singleton is irredundant
  $K^+ \leftarrow K^+ \cup$ **MaxIrredOutrankingChoices** $(Y_0, K^+, Hist)$

def MaxIrredOutrankingChoices (**In:** Y_i +irredundant, K_i^+; **Out:** K_{i+1}^+):
  if $(Y_i - X) - N^+(Y_i) = \emptyset$:
    $K_{i+1}^+ \leftarrow K_i^+ \cup Y_i$ # Y_i is outranking
  else:
    $K_{i+1}^+ \leftarrow K_i^+$ # initialise the result
    for $[x \in X - Y_i : N_{Y_i}^+[x] \neq \emptyset]$: # add all +irredundant actions
      $Y_{i+1} \leftarrow Y_i \cup \{x\}$
      if $Y_{i+1} \notin Hist$:
        $K_{i+1}^+ \leftarrow K_{i+1}^+ \cup$ **MaxIrredOutrankingChoices** (Y_{i+1}, K_{i+1}^+):
        $Hist \leftarrow Hist \cup \{Y_{i+1}\}$
  return K_{i+1}^+

Proof. The algorithm starts with an empty history and an empty set of minimal outranking choices. The procedure **MaxIrredOutrankingChoices** then collects all minimal outranking choices that may be reached in turn from each initial single choice $Y_0 = \{x\}, \forall x \in X$

The call invariants of iteration i are that the current choice Y_i is +irredundant and K_i^+ contains the minimal outranking choices collected so far.

If Y_i is -irredundant, then $Y_{i+1} = Y_i \cup \{x\}$ is constructed only if $N_{Y_i}^+[x] \neq \emptyset$, i.e. in case x is a +irredundant action with respect to current choice Y_i. Y_{i+1} remains therefore +irredundant. As each Y_0 is in turn +irredundant, the algorithm walks only on paths of the +irredundant hypergraph.

Let us suppose that at iteration i, K_i^+ is either empty or contains only minimal outranking choices. Two situations may happen. First, the current choice Y_i is outranking and we have found a maximal irredundant, i.e. a minimal outranking choice, and we add it to the current set K_i^+. In the second case, we gather all minimal outranking choices from the union of the current choice Y_i with all possible +irredundant actions, i.e. such that $N_{Y_i}^+[x] \neq \emptyset$. This way, K_{i+1}^+ can only contain minimal outranking choices or stay empty. As we start with an empty initial collection K_0^+, it is verified that in the end K^+ may only contain minimal outranking choices.

Finally, that the algorithm collects all existing minimal outranking choices in $G^{\mathcal{L}}$ follows from the fact that the +irredundant choice graph is strongly connected. Starting in turn from each single choice, the algorithm walks necessarily through all +irredundant choices existing in $H(\mathcal{P}(X), +irredundent)$. In order to avoid visiting the same +irredundant choices several times in turn from each member single choice, we keep a history of visited +irredundant choices, and only proceed recursively with the next choice Y_{i+1} in case it has not been visited already before. □

The same algorithm delivers again the minimal outranked choices when replacing the outranked with the outranking neighbourhoods. This way, we only walk on -irredundant choices and collect all minimal outranked choices instead.

2.2 Complexity and performance

The problem of finding a minimal outranking or outranked choice of a certain cardinality k, is known to be NP-complete, so that there is little hope to find efficient algorithms for enumerating all minimal outranking or outranked choices in general digraphs of high orders.

Indeed, the complexity is directly linked to the size of the P-choice graphs. In case a digraph $G^{\mathcal{L}}$ is empty, only the greedy choice will actually be a outranking choice. The outranking choice graph reduces here to a single node and algorithm 1 will deliver immediately this unique possible solution. As every possible choice in $\mathcal{P}(X)$ will be irredundant, the corresponding $\pm$-irredundant choice graph will be of order $2^n - 1$ (where n is the order of G) and of size $(2^n - 1)^2 - (2^n - 1)$. Algorithm 2 therefore rapidly gets totally inefficient.

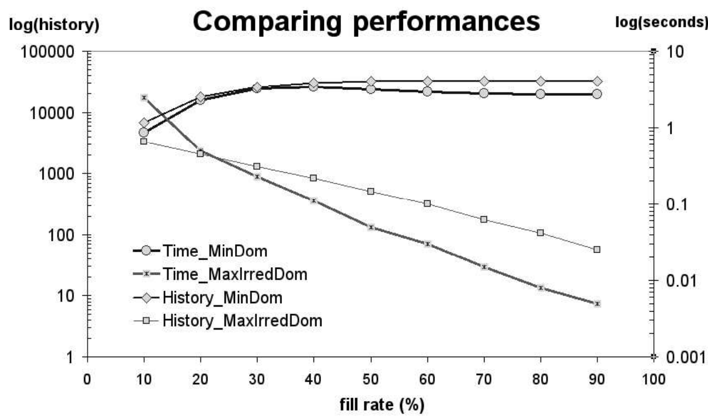
Similarly, in case $G^{\mathcal{L}}$ is complete, i.e. G is a complete graph K_n, the irredundant choice graph reduces to n isolated single choices. This time, algorithm 2 delivers immediatley the n solutions, whereas the corresponding outranking choice graph is again of order $2^n - 1$ and so of huge size $(2^n - 1)^2 - (2^n - 1)$. Similarly, algorithm 1 this time is totally inefficient.

A stated before we are in fact solely interested in connected digraphs where the second algorithm is more efficient in general, except for very low fill rates (see figure (1). We have implemented both algorithms in the Python language (version 2.4) using the optimized inbuilt `set` class, which delivers constant time access to members of sets (independent of the cardinalities), and which offers optimized set operators like union, intersection, and difference with linear time in the cardinality of the operands. In figure (1) we have illustrated run time statistics for random digraphs of order 15 with fill rates varying from 10 to 90%.

It is obvious that the `MaxIrredOutrankingChoices` algorithm is doing much better except for very low fill rate below 15 %.

Let us now consider a special kind of outranking and outranked choices, namely those where the chosen ations are incomparable with repect to the S relation.

Figure 1: Run time statistics for randomly filled connected digraphs of order 15



2.3 Choice graph traversal algorithms for kernel enumerating

We have seen in the first section, that the independence property is computed from the $\mathcal{L}$-false part of $\tilde{S}$. In order to implement path algorithms in the corresponding independent choice graph, we cannot, as usual rely on the false by failure principle, i.e. the complement of the neighbourhoods, for representing independence. We need to introduce the logically positive concept of *disconnects*.

Definition 10 (Disconnects)**.**
Let $G^{\mathcal{L}}$ be an $\mathcal{L}$-irreflexive digraph. We call *disconnect* of a node x, denoted $D(x) = \{y \in X : (\tilde{S}(y,x) < 0) \;\vee\; (\tilde{S}(x,y) < 0)\}$, the set of nodes *disconnected* from x. We call *disconnect* of a choice Y, the intersection of disconnects of the members of Y:

$$D(Y) = \bigcap_{x \in Y} D(x).$$

Proposition 6. *A choice Y in $G^{\mathcal{L}}$ is an outranking (respectively outranked) kernel if and only if:*

$$\begin{array}{rcll} & & Y \subseteq D(Y) & \textit{(independent)} \\ \forall x \notin Y & : & N^-(x) \cap Y \neq \emptyset & \textit{(outranking)} \\ \textit{(resp. } \forall x \notin Y & : & N^+(x) \cap Y \neq \emptyset & \textit{(resp. outranked))} \end{array}$$

Proof. It is readily seen that a choice Y is indeed independent if and only if the disconnects of the choice members contain the otherwise chosen actions. Similarly, a choice Y is outranking (resp. outranked) if and only if all not members of the choice are in the respective choice neighbourhood. □

2.3.1 Reducing outranking choices

Algorithm 3 (Enumerating outranking choices 2)**.**

$Y_0 \leftarrow X$ # start with the greedy choice
$K^+ \leftarrow$ **MinOutrankingKernels** (Y_0)

def MinOutrankingKernels (**In:** Y outranking; **Out:** K^+)
  if $Y \subseteq D(Y)$:
    $K^+ \leftarrow Y$ # Y is independent
  else:
    $K^+ \leftarrow \emptyset$
    for $[x \in Y : \; N^+_Y[x] = \emptyset]$: # Retract in turn all redundant nodes
      $Y_1 \;\leftarrow\; Y - \{x\}$ # $Y1$ remains outranking !
      $K^+ \;\leftarrow\; K^+ \cup$ **MinOutrankingKernels** (Y_1)
  return K^+

Proof. Similar in its design to algorithm 1, this algorithm starts again with the greedy choice $Y = X$ which is always outranking by convention and an empty set of minimal outranking kernels. The procedure **MinOutrankingKernels** collects all independent outranking choices that may be reached from this initial outranking choice Y.

The call invariants of iteration i are that the choice Y_i is outranking and K^+_i is a set of outranking kernels collected so far.

If Y_i is outranking, then $Y_{i+1} = Y_i - \{x\}$ is constructed only if $N^+_{Y_i}[x] = \emptyset$, i.e. when x is a +irredundant action, so that Y_{i+1} remains outranking. If no more +irredundant actions may be found, the procedure stops the walk. As $Y_0 = X$ is outranking, the algorithm only walks on paths of the outranking choice graph.

Let us suppose that at iteration i, K^+_i contains only outranking kernels. Two situations may happen. Either the current choice Y_i is independent or all redundant actions have been removed in turn. In the first case, we are in the presence of an outranking kernel which is added to to the current set K^+_i. In the second case, all outranking kernels potentially reached when reducing the current choice are first added up in a local result K^+ to be at the end added up to K^+_i. This way, K^+_{i+1} can only contain outranking kernels. As we start with an empty initial collection K^+_0, it is verified that in the end K^+ may only contain minimal outranking choices.

Finally, that we algorithm collects all existing independent outranking choices in $G^{\mathcal{L}}$ follows from the fact that the outranking chice graph is strongly connected and that therefore, starting from the greedy choice X, the algorithm walks necessarily through all outranking choices in $G^{\mathcal{L}}$. □

2.3.2 Extending independent choices

Algorithm 4 (Extending independent choices: variant 1)**.**

$K^+ \leftarrow \emptyset$ # initialise the result
for $x \in X$:
  $Y \leftarrow \{x\}$ # each singleton is independent
  $K^+ \leftarrow K^+ \cup$ **MaxIndOutrankingKernels** (Y, K^+)

def MaxIndOutrankingKernels(**In:** Y independent, K_0^+; **Out:** K^+):
  if $N^+(Y) - (Y - X) = \emptyset$:
    $K^+ \leftarrow K_0^+ \cup Y$ # Y is outranking
  else: # try adding all independent singletons
    $K^+ \leftarrow K_0^+$ # initialise the result
    for $[x \in X - Y : Y - \{x\} \subseteq D(x)]$:
      $Y_1 \leftarrow Y \cup \{x\}$ # Y_1 remains independent !
      $K^+ \leftarrow K^+ \cup$ **MaxIndOutrankingKernels** (Y_1, K^+)
  return K^+

Before going to prove algorithm 4, we may notice that the independence property in the recursive call invariant here, contrary to the $\pm$-irredundancy properties, is a non oriented concept. This allows to enumerate in the same run, both the outranking and the outranked kernels.

2.3.3 Dominant and outranked kernels in the same run

Algorithm 5 (Extending independent choices: variant 2)**.**

global $Hist$
$Hist \leftarrow \emptyset$ # initialise the history
$K^+ \leftarrow \emptyset$ # initialise the outranking result
$K^- \leftarrow \emptyset$ # initialise the outranked result
for $x \in X$:
  $Y \leftarrow \{x\}$
  $(K^+, K^-) \leftarrow (K^+, K^-) \cup$ **AllKernels** $(Y$**,** $(K^+, K^-))$

def AllKernels (**In:** Y independent, (K_0^+, K_0^-); **Out:** (K^+, K^-)):
  if $N^+(Y) - (Y - X) = \emptyset$:
    $K^+ \leftarrow K_0^+ \cup Y$ # Y is outranking
  if $N^-(Y) - (Y - X) = \emptyset$:

$K^- \leftarrow K_0^- \cup Y$ # Y is outranked
try adding all independent singletons
$(K^+, K^-) \leftarrow (K_0^+, K_0^-)$
for $[x \in D(Y)]$:
$Y_1 \leftarrow Y \cup \{x\}$
if $Y_1 \notin Hist$:
$(K^+, K^-) \leftarrow (K^+, K^-) \cup$ **AllKernels** $(Y_1, (K^+, K^-))$
$Hist \leftarrow Hist \cup Y_1$
return (K^+, K^-)

Proof. The algorithm starts with an empty history and empty sets of outranking and outranked kernels. The procedure **AllKernels** then collects all outranking and outranked kernels that may be reached in turn from each initial single choice $Y_0 = \{x\}, \forall x \in X$.

The call invariants of procedure **AllKernels** are that the current choice Y_i is independent, and that the current set K_i^+ (resp. K_i^-) of results contains the outranking (respectively outranked) kernels collected so far.

If Y_i is independent, then $Y_{i+1} = Y_i \cup \{x\}$ is constructed only if $x \in D(Y_i)$, i.e. in case Y_{i+1} remains independent. As each Y_0 is in turn independent by convention, the algorithm walks only on paths of the independent choice graph.

Let us suppose that at recursive call i, K_i^+ and K_i^- are either empty or contain only outranking or outranked kernels. Three situations may happen. First, the current choice Y_i is outranking and we have found a new outranking kernel that we add to the current set K_i^+. In the second case, the current choice Y_i is outranked and we have found a new outranked kernel that we add again to the current set K_i^-. Thirdly, we gather all outranking and outranked kernels from the union of the current choice Y_i with all possible actions x contained in its disconnect. This way, K_{i+1}^+ and K_{i+1}^- can only contain outranking, respectively outranked kernels or stay empty. As we start with empty initial collections K_0^+ and K_0^-, it is verified that in the end K^+, respectively K^-, if not empty, may only contain outranking, respectively outranked, kernels.

Finally, that the algorithm collects all existing outranking and outranked kernels in $G^{\mathcal{L}}$ follows from the fact that the independent hypergraph is strongly connected. Starting in turn from each single choice, the algorithm walks necessarily through all independent choices existing in $G^{\mathcal{L}}$. In order to avoid visiting the same independent choices several times in turn from each member single choice, we keep a history of visited independent choices, and only proceed recursively with the next choice Y_{i+1} in case it has not been visited before. □

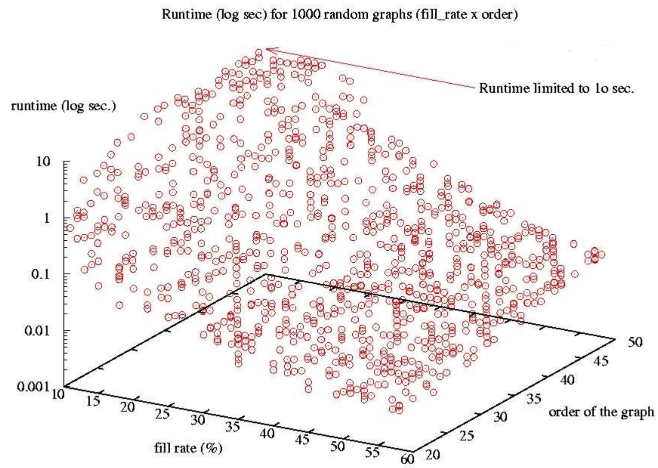
Figure 2: Run time statistics for AllKernels procedure (Algorithm 5)

2.4 Complexity and computational performance

In figure 2 we show run times statistics for kernel extractions from randomly filled digraphs of order 30. Similar to the previous statistics, we find that the extraction of kernels is computationally easy (run times less than a second) when the fill rate is 20% and more. The performance is again directly related to the order of the independent choice graph. Indeed, the higher the fill rate, the lower is the order of this choice graph. With a fill rate of 50% for instance, we observe an average of only 200 independent choices. We may collect on this low order independent choice graph the outranking and outranked kernels in an average of 15 milliseconds on a standard desktop PC.

This run time performance is even better supported in general (see figure 3) when considering that almost all digraphs of order n contain only kernels such that $C_n - 1.43 \leq |K| \leq C_n + 2.11$ where $C_n = ln(n) - ln(ln(n))$ (Tomescu [20]). For a ramdomly filled digraph of order 900 and 50% fill rate, we may thus observe kernels of average cardinalities of 7. Thus we are able to extract in less than a minute all kernels from digraphs of orders up to 900 and a fill rate of 50% and more, under the condition of disposing of a sufficiently large CPU memory. This general performance is most satisfactory, as the particular outranking graphs we are interested in generally represent more or less transitive weak orderings. As empiric studies of random outranking digraphs is confirming, the corresponding digraphs show fill rates always superior to 50%. Nevertheless some digraphs, even of modest order (less than 30), may represent difficult instances. Indeed, as shown in figure 3, where we have articially limited the run time to 10 seconds, a brutal

Figure 3: General performance of Algorithm 5



combinatorial explosion appears with digraphs of very low fill rate. Here we may easily observe independent choice graphs of huge exponential size coupled with kernels of cardinalities up to $n/2$. This definitely limits the practical performance for extracting all kernels from these kinds of digraphs.

But the independent choice graph traversal approach is not the only possible strategy for computing kernels in a digraph. Very recently, Alain Hertz[0] has proposed a pivoting algorithm which, starting from an arbitrary initial maximal independent choice, visits directly all other existing maximal indpendent sets in the digraph. This algorithm belongs to the famaily of reverse searching algorithms such as the simplex algorithm in linear algebra. The pivoting from one maximal independent choice to the other is done in a polynomial $\mathcal{O}(n)$ step, so that performances in fact only depend on the actual number of kernels existing in the digraph. Even if this last algorithm is not as efficient as the Allkernels algorithm for dense digraps of large orders, it however delivers all kernels for difficult digraphs such as cordless n-circuits, and n-paths.

All the preceding discussion only concerns the computation of kernels in the associated median cut crips digraph. In the next section we propose an algebraic approach to the same problem via $\mathcal{L}$-valued membership characterisations of choices, which will deliver the necessary algorithms for solving the general $\mathcal{L}$-valued case.

[0]Private communication, April 2006

3 Algebraic approach

3.1 $\mathcal{L}$-characterisation of choice classes

In the previous sections we have worked with different kinds of choices, namely outranking, outranked, independent, $\pm$-irredundant ones. Similarly to the $\mathcal{L}$-characterisation of the digraph, we may now define a $\mathcal{L}$-valued characterisation of these kinds or classes on the power set $\mathcal{P}(X)$ of all possible choices we may define in $G^{\mathcal{L}}$.

As these classes are all defined with logical conditions applied on $\mathcal{L}$-valued binary outranking statements, we first need to extend the $\mathcal{L}$ evaluation domain to well formed logical expressions.

Definition 11 (Well formed logical expressions)**.**
Let $\mathcal{G}$ denote a set of ground atomic logical statements. We define inductively the set $\mathcal{E}$ of well formed logical expressions in the following way:

1. $\forall p \in \mathcal{G}$ we have $p \in \mathcal{E}$;
2. $\forall x, y \in \mathcal{E}$ we have $(x \vee y) \in \mathcal{E}$, $(x \wedge y) \in \mathcal{E}$, and $\neg x \in \mathcal{E}$
3. all $p \in \mathcal{E}$ result of finite construction.

In order to avoid any problem with precedence of operators, we shall always use brackets to delimit the scope of the logical operators $\max, \min$ and $\neg$ in an expression. Here our ground atomic logical expressions are the binary outranking assertions $x \,\mathrm{S}\, y$ of the given digrah $G^{\mathcal{L}}(X, \tilde{S})$. Our well formed logical expressions concern formulas involving these binary outranking assertions.

Now, every well formed expression may be evaluted in the $\mathcal{L}$-valued credibility domain as follows:

Definition 12 ($\mathcal{L}$-valued logical expressions)**.**
Let $\mathcal{E}$ denote a set of well formed logical expressions. $\tilde{\mathcal{E}} : \ \mathcal{E} \rightarrow \mathcal{L}$ gives the $\mathcal{L}$-valued credidility of each well formed logical expression as follows:

1. $\forall p \in \mathcal{G}$, its credibility $\tilde{\mathcal{E}}(p) \in \mathcal{L}$ is given.
2. $\forall x \in \mathcal{E}$ we have $\tilde{\mathcal{E}}(\neg x) = -\tilde{\mathcal{E}}(x)$
3. $\forall x, \in \mathcal{E}$ we have $\tilde{\mathcal{E}}(x \vee y) = \max\big(\tilde{\mathcal{E}}(x), \tilde{\mathcal{E}}(y)\big)$
4. $\forall x, \in \mathcal{E}$ we have $\tilde{\mathcal{E}}(x \wedge y) = \min\big(\tilde{\mathcal{E}}(x), \tilde{\mathcal{E}}(y)\big)$

As the atomic outranking assertions are evaluated in the given digraph $G^{\mathcal{L}}(X, \tilde{S})$, we are now able to evaluate any well formed logical expression involving these evaluations $\tilde{S}(x, y)$. We start by defining the degree of irredundance of a choice in $G^{\mathcal{L}}$.

Definition 13 ($\mathcal{L}$-$\pm$-irredundance of choices)**.**
Let $G^{\mathcal{L}}(X, \tilde{S})$ be a $\mathcal{L}$-valued digraph where $\mathcal{L} = \{-m, \ldots, 0, \ldots, m\}$. The degree of +-*irredundance* of action x with respect to choice Y in $G^{\mathcal{L}}$ is given by:

$$\Delta_Y^{\text{+irr}}(x) = \begin{cases} m & \text{if} \quad Y = \{x\}, \\ \max_{(z,y) \in X \times Y - \{x\}} \min\big(\, \tilde{S}(x,z), -\tilde{S}(y,z) \,\big) & \text{otherwise}. \end{cases} \tag{3}$$

Similarly, the degree of $-$-*outranked irredundance* of action x with respect to choice Y in $G^{\mathcal{L}}$ is given by:

$$\Delta_Y^{\text{-irr}}(x) = \begin{cases} m & \text{if} \quad Y = \{x\}, \\ \max_{(z,y) \in X \times Y - \{x\}} \min\big(\, \tilde{S}(z,x), -\tilde{S}(z,y) \,\big) & \text{otherwise}. \end{cases} \tag{4}$$

The degree of +*irredundance* of choice Y in $G^{\mathcal{L}}$ is given by:

$$\Delta^{\text{+irr}}(Y) = \min_{x \in Y} \Delta_Y^{\text{+irr}}(x) \tag{5}$$

The degree of -*irredundance* of choice Y in $G^{\mathcal{L}}$ is given by:

$$\Delta^{\text{-irr}}(Y) = \min_{x \in Y} \Delta_Y^{\text{-irr}}(x) \tag{6}$$

Proposition 7.
Y in $G^{\mathcal{L}}$ is a +irredundant outranking (resp. -irredundant) choice if and only if $\Delta^{\text{+irr}}(Y) > 0$ (resp. $\Delta^{\text{-irr}}(Y) > 0$).

Proof.
(⇒) Suppose $\Delta^{\text{+irr}}(Y) < 0$. Then $\exists x \in Y$ such that $\Delta_Y^{\text{+irr}}(x) < 0$. This implies that $Y \subset X$ and $\forall (z,y) \in X \times Y - \{x\}$ we have $\min\big(\, \tilde{S}(x,z), -\tilde{S}(y,z)\big) < 0$. In other terms: $\forall z \in N^{+}[x] : \exists y \in Y - \{x\}$ such that $z \in N^{+}[y]$. Hence x is redundant and Y cannot be +irredundant.

(⇐) Let us suppose the otherway round that x in choice Y is redundant. This implies that $N^{Y} + [x] = \emptyset$. In other terms: $N^{[}x] - N^{+}[Y - \{x\}] = \emptyset$. This is exactly the case when for all $z \in X$ such that $\tilde{S}(x,z) > 0$, we find a $y \in Y - \{x\}$ such that $\tilde{S}(y,z) > 0$. In this case $\max_{(z,y) \in X \times Y - \{x\}}(\tilde{S}(z,x), -\tilde{S}(z,y)) < 0$ and $\Delta_Y^{\text{+irr}}(x) < 0$.

A same development applies for the outranked case. □

Definition 14 ($\mathcal{L}$-Qualification of choices)**.**
Let $G^{\mathcal{L}}(X, \tilde{S})$ be a $\mathcal{L}$-valued digraph where $\mathcal{L} = \{-m, \ldots, 0, \ldots, m\}$. The *degree of outrankingness* of a choice Y in $G^{\mathcal{L}}$ is given by:

$$\Delta^{\text{dom}}(Y) = \begin{cases} m & \text{if} \quad Y = X, \\ \min_{x \notin Y} \max_{y \in Y} \big(\tilde{S}(y,x)\big) & \text{otherwise.} \end{cases} \tag{7}$$

The *degree of outrankedness* of a choice Y in $G^{\mathcal{L}}$ is given by:

$$\Delta^{\text{abs}}(Y) = \begin{cases} m & \text{if} \quad Y = X, \\ \min_{x \notin Y} \max_{y \in Y} \big(\tilde{S}(x,y)\big) & \text{otherwise.} \end{cases} \tag{8}$$

The *degree of independence* of a choice Y in $G^{\mathcal{L}}$ is given :

$$\Delta^{\text{ind}}(Y) = \begin{cases} m & \text{if} \quad Y = \{x\}, \\ \min_{y \in Y}^{y \neq x} \min_{x \in Y} \big(-\tilde{S}(x,y)\big) & \text{otherwise.} \end{cases} \tag{9}$$

Proposition 8.
Let $G^{\mathcal{L}}(X, \tilde{S})$ be an $\mathcal{L}$-valued outranking graph.

1. *Y in $G^{\mathcal{L}}$ is an independent choice if and only if $\Delta^{ind}(Y) > 0$.*
2. *Y in $G^{\mathcal{L}}$ is a outranking (resp. outranked) choice if and only if $\Delta^{dom}(Y) > 0$ (resp. $\Delta^{abs}(Y) > 0$).*

Proof. (1) Immediate from definition (7) which states that a choice Y is indeed independent if and only if $\tilde{S}(x,y) < 0$ for all $x, y \in Y$.

(2) similarly, follows immediately from definition (1), as a choice Y is outranking (resp. outranked) if and only if $\forall x \in Y : \exists y \in Y$ such that $\tilde{S}(y,x) > 0$ (resp. $\tilde{S}(y,x) > 0$. □

Corollary 1. Let $G^{\mathcal{L}}(X, \tilde{S})$ be an $\mathcal{L}$-valued outranking graph and $G(X, S)$ its associated strict median cut crisp digraph. The minimal outranking (resp. outranked) choices of $G^{\mathcal{L}}$ correspond to the minimal outranking (resp. outranked) choices of G.

Proof. Y in $G^{\mathcal{L}}$ is a minimal outranking (resp. outranked) choice if and only if $\Delta^{\text{+irr}}(Y) > 0$ and $\Delta^{\text{dom}}(Y) > 0$ (resp. $\Delta^{\text{-irr}}(Y) > 0$ and $\Delta\text{abs}(Y) > 0$). □

This important result from an operational point of view allows to determine the $\mathcal{L}$-valued minimal outranking (resp. outranked) choices in a $\mathcal{L}$-valued digraph $G^{\mathcal{L}}$ by – first, computing the minimal outranking (resp. outranked) crisp choices in the associated strict median cut digraph G, and – secondly, computing their respective $\mathcal{L}$-qualifications.

Corollary 2 (Kitainik 1993)**.** The outranking (resp. outranked) kernels of $G^{\mathcal{L}}$ correspond to the outranking (resp. outranked) kernels of G.

Proof. Let $G^{\mathcal{L}}(X, \tilde{S})$ be an $\mathcal{L}$-valued outranking graph. Y in $G^{\mathcal{L}}$ is a outranking (resp. outranked) kernel if and only if $\Delta^{\text{ind}}(Y) > 0$ and $\Delta^{\text{dom}}(Y) > 0$ (resp. $\Delta\text{abs}(Y) > 0$). ☐

Again, this result allows us to determine all outranking or outranked kernels in a $\mathcal{L}$-valued digraph $G^{\mathcal{L}}$ by – first, extracting all crisp kernels from the associated median cut crisp graph G and, – secondly, directly computing their corresponding $\mathcal{L}$-qualifications.

3.2 The kernel equation system

Definition 15 ($\mathcal{L}$-characterisation of choices)**.**
We characterise Y with the help of a $\mathcal{L}$-valued function $\tilde{Y} : X \to \mathcal{L}$ where $x \in Y \Leftrightarrow \tilde{Y}(x) > 0,\ \forall x \in X$.

In example (1), $\tilde{Y}(a) = -6$, $\tilde{Y}(b) = \mathbf{6}$, $\tilde{Y}(c) = -6$, $\tilde{Y}(d) = \mathbf{10}$, $\tilde{Y}(e) = \mathbf{9}$ characterises the choice $Y = \{b, d, e\}$, whereas $\tilde{Y}(a) = \mathbf{6}$, $\tilde{Y}(b) = -6$, $\tilde{Y}(c) = \mathbf{6}$, $\tilde{Y}(d) = -10$, $\tilde{Y}(e) = -9$ characterises the choice $Y = \{a, c\}$.

Definition 16 (The kernel equation system)**.**
We call outranking kernel equation system the following set of equations:

$$(\tilde{Y} \circ \tilde{S})(x) = \max_{y \in X}^{y \neq x} \min\big(\tilde{Y}(y), \tilde{S}(y, x)\big) = -\tilde{Y}(x),\ \forall x \in X. \qquad (10)$$

We call outranked kernel equation system the following set of equations:

$$(\tilde{Y} \circ \tilde{S})(x) = \max_{y \in X}^{y \neq x} \min\big(\tilde{Y}(y), \tilde{S}(x, y)\big) = -\tilde{Y}(x),\ \forall x \in X. \qquad (11)$$

The name given to both these $\mathcal{L}$-valued equation systems is motivated by the following result we observe in the crisp Booelan setting.

Theorem 1 (Berge 1958)**.**
Let $G^{\mathcal{L}}(X, \tilde{S})$ be evaluated in a bi-valued domain $\mathcal{L} = \{-1, 1\}$. A choice Y in $G^{\mathcal{L}}$ is a outranking (resp. outranked) kernel if and only if its associated $\{-1, 1\}$-valued characteristic vector $\tilde{Y}$ is a solution of kernel equation system (10) (resp. (11)).

Proof. ($\Rightarrow$) Let us suppose that $\tilde{Y}$ characterises a outranking kernel in $G^{\mathcal{L}}$. By independence of Y we have $\tilde{Y}(y) = 1 \Rightarrow y \in Y \Rightarrow \min_{x \neq y \in Y} \tilde{S}(y, x) = -1$. And, by the outrankingness quality of Y, we have $\tilde{Y}(y) = -1 \Rightarrow y \notin Y \Rightarrow \max_{x \in Y} \tilde{S}(y, x) = 1$. Combining both cases we see that $\tilde{Y}$ indeed verifies equation system (10).

($\Leftarrow$) If $\tilde{Y}$ is a solution of kernel equation system (10),
$y \in Y \Rightarrow \tilde{Y} = 1 \Rightarrow \min_{x \neq y \in Y} \tilde{S}(x, y) = -1$ and Y is independent. Similarly, $y \notin Y$ $\Rightarrow \tilde{Y}(y) = -1 \Rightarrow \max_{x \neq y \in Y} \tilde{S}(y, x) = 1$ and Y is indeed a outranking choice. If we combine both cases, Y is certainly a outranking kernel.

A same argument applies canonically for outranked kernels. □

In the general $\mathcal{L}$-valued case, the correspondence between the solutions of the kernel equation systems and $\mathcal{L}$-valued kernels is not so immediate.

Indeed, we have to be careful with potential $\mathcal{L}$-undeterminedness. In fact the only $\mathcal{L}$-characterisations we may accept are those that determine a complete choice.

Definition 17 ($\mathcal{L}$-determined choices)**.**
Let $\tilde{Y}$ represent a $\mathcal{L}$-characterisation of a choice Y in $G^{\mathcal{L}}$. We call $\tilde{Y}$ $\mathcal{L}$-determined if $\tilde{Y}(x) \neq 0$ for all $x \in X$.

Furthermore, certain $\mathcal{L}$-characterisations, despite being different in values, characterise in fact a same choice. To cope with this phenomena, we have to introduce the following congruence relation on $\mathcal{Y}$, the set of possible $\mathcal{L}$-characterisations of choices in $G^{\mathcal{L}}$.

Definition 18 (Congruence classes of $\mathcal{L}$-characterisations)**.**
We say that two $\mathcal{L}$-characterisations $\tilde{Y_1}$ and $\tilde{Y_1}$ of kernels in $G^{\mathcal{L}}$ are *non contradictory*, denoted $\tilde{Y_1} \cong \tilde{Y_2}$ if and only if $\tilde{Y_1}(x) > 0 \Leftrightarrow \tilde{Y_2}(x) > 0$ and $\tilde{Y_1}(x) > 0 \Leftrightarrow \tilde{Y_2}(x) > 0$. Every choice Y in $G^{\mathcal{L}}$ determines a congruence class of non contradictory $\mathcal{L}$-characterisations denoted $\mathcal{Y}_{/\cong Y}$.

Definition 19 (Sharpness of $\mathcal{L}$-characterisations)**.**
Let $\tilde{Y_1}, \tilde{Y_2} \in \mathcal{Y}_{/\cong Y}$ characterise a choice Y in $G^{\mathcal{L}}$. We say that $\tilde{Y_1}$ is sharper than $\tilde{Y_2}$, denoted $\tilde{Y_1} \succcurlyeq \tilde{Y_2}$ if and only if for all $x \in X$, either $\tilde{Y_1}(x) \leq \tilde{Y_2}(x) \leq 0$, or $0 \leq \tilde{Y_2}(x) \leq \tilde{Y_1}(x)$.

The sharpness relation $\succcurlyeq$ determines a partial order on $\mathcal{Y}$, the set of possible $\mathcal{L}$-characterisations of choices in $G^{\mathcal{L}}$ (see [5]). The all median valued vector $\tilde{Y_0}(x) = 0, \forall x \in X$ acts as bottom, the least sharpest characterisation and all 2^n crisp, i.e. $\{-m, m\}$-valued choice characterisations give the sharpest possible characterisations.

Theorem 2 (Bisdorff, Pirlot, Roubens, 2005)**.** A choice Y is a outranking (resp. outranked) kernel in $G^{\mathcal{L}}$ if and only if there exists a corresponding $\mathcal{L}$-valued characteristic vector $\tilde{Y}$ that is a maximal sharp $\mathcal{L}$-determined solution of the kernel equation system (10) (resp. (11)).

Proof. ($\Leftarrow$) If $\tilde{Y}$ is a maximal sharp and $\mathcal{L}$-determined solution of equation system (10), then the so characterised choice Y will be independent and outranking as a corollary of theorem (1).

($\Rightarrow$) If Y is a outranking kernel in $G^{\mathcal{L}}$ we show that there exists a unique solution $\tilde{Y} \in \mathcal{Y}_{/\cong Y}$ of the fixpoint equation :

$$\mathcal{T}(\tilde{Y}) = -(\tilde{Y} \circ \tilde{S}) = \tilde{Y} \tag{12}$$

that is a maximal sharp and $\mathcal{L}$-determined solution of equation system (10).

Indeed, it is readily seen that the fixpoints of equation (12) verify in fact the kernel equation system (10).

Transformation $\mathcal{T}$ gives furthermore a non-contradictory transformation of kernel characterisations, i.e. $\tilde{Y} \in \mathcal{Y}_{/\cong Y} \Rightarrow \mathcal{T}(\tilde{Y}) \in \mathcal{Y}_{/\cong Y}$. Indeed, $y \in Y \Rightarrow \tilde{S}(y,x) < 0$ so that $\forall x \in Y$, $\min(\tilde{Y}(x), \tilde{S}(x,y)) = \tilde{S}(x,y) < 0$, and, $\forall x \notin Y$, $\min(\tilde{Y}(x), \tilde{S}(x,y)) \leq \tilde{Y}(x) < 0$. The combination of both cases shows that $\mathcal{T}(\tilde{Y})(y) > 0$. Similarly, $x \notin Y \Rightarrow \exists y \in Y : \tilde{S}(y,x) > 0$. For such an y, $\min(\tilde{Y}(x), \tilde{S}(x,y)) > 0$ and hence $\mathcal{T}(\tilde{Y}) < 0$.

We may also show that the transformation $\mathcal{T}$ is isotone with respect to the sharpness ordering $\succcurlyeq$, i.e. if $\tilde{Y}_1, \tilde{Y}_2 \in \mathcal{Y}_{/\cong Y}$ are such that $\tilde{Y}_1 \succcurlyeq \tilde{Y}_2$ then $\mathcal{T}(\tilde{Y}_1) \succcurlyeq \mathcal{T}(\tilde{Y}_2)$. Indeed, $y \in Y \Rightarrow \tilde{Y}_1(y) > \tilde{Y}_2(y) \Rightarrow \mathcal{T}(\tilde{Y}_1)(y) > \mathcal{T}(\tilde{Y}_2)(y)$, and $y \notin Y \Rightarrow \tilde{Y}_1(y) < \tilde{Y}_2(y) \Rightarrow \mathcal{T}(\tilde{Y}_1)(y) < \mathcal{T}(\tilde{Y}_2)(y)$ since the functions $\max$ and $\min$ are non decreasing.

If we start now the resolution of the fixpoint equation with $\tilde{Y}_0(x) = m$ when $x \in Y$, and $\tilde{Y}_0(x) = -m$ when $x \notin Y$, i.e. the maximal possible sharp characterisation, we necessarily get $\tilde{Y}_i \succcurlyeq \mathcal{T}(\tilde{Y}_{i-1})$ for $i = 1, 2, \ldots$. As m is a finite integer, there exists a finite number $n \leq n(m-1)$ such that $\tilde{Y}_n = \mathcal{T}(\tilde{Y}_n)$.

This fixpoint solution $\tilde{Y}_n$ is unique, $\mathcal{L}$-determined and maximal sharp.

The outranked case is canonically obtained by simply reversing the $\tilde{S}$ relation. □

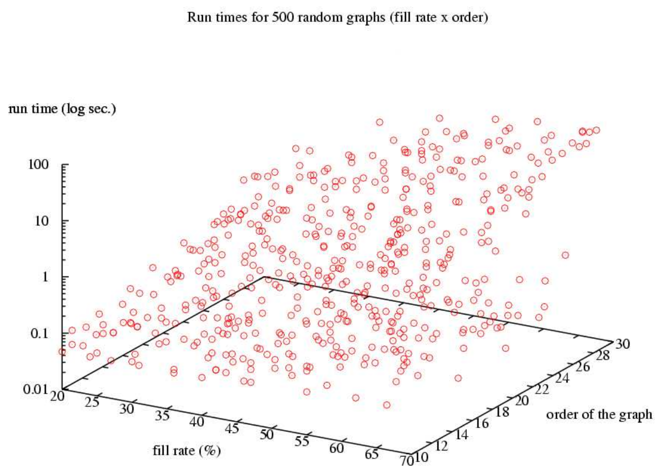
The last theorem gives us the possibility to find a $\mathcal{L}$-characterisation of a outranking (resp. outranked) kernel under the condition that we already know the associated strict median cut choice. Let us now turn our attention to direct solving techniques for the kernel equation systems.

3.3 Solving the kernel equation systems

3.3.1 Smart enumeration with a finite domain solver

It is possible to directly enumerate all maximal sharp solutions from the $\mathcal{L}$-valued kernel equations systems with the help of a finite domain solver as provided by some Prolog pro-

Figure 4: Average performance using the GNU-Prolog FD solver



gramming environments such as GNU-Prolog [10; 11] or the commercial Prolog software CHIP. Implementation details of such a solving approach may be found in Bisdorff [4].

In Figure 4, we show average performance using the GNU-Prolog FD solver. Contrary to our `AllKernels` Python implementation, better performances are obtained here with smaller fill rates. This is due to the size of the arc-constraints graph which is indeed proportional to the actual size of treated outranking digraph. The sparser the graph, the smaller the constraints graph, the quicker the propagation algorithm on the arc-constraint will help enumerating all kernels in the graph.

However, direct enumeration in a bipolar-valued characteristic domain is very inefficient. It quickly appeared that fixpoint approaches are much more efficient (see Bisdorff [5]).

3.3.2 Fixpoint approaches

Before tackling the general $\mathcal{L}$-valued case, we may consider the following early result.

Theorem 3.
Let $G^{\mathcal{L}}(X, \tilde{S})$ be a bipolar-valued outranking digraph such that their exists a unique kernel K in $G^{\mathcal{L}}$ with the associated maximal sharp and $\mathcal{L}$-determined $\tilde{K}$ characterisation. Let $\mathcal{T}^2 : \mathcal{Y} \rightarrow \mathcal{Y}$ be the following dual transformation of $\mathcal{L}$-valued kernel characterisations:

$$\mathcal{T}^2(\tilde{Y}) = -\big(- (\tilde{Y} \circ \tilde{S}) \circ \tilde{S}\big). \tag{13}$$

With $\tilde{Y}_0(x) = -1$ for all $x \in X$, the iteration $\tilde{Y}_i = \mathcal{T}^2(\tilde{Y}_{i-1})$ for $i = 1, 2, \ldots$ converges to the fixpoint $\tilde{K} = \mathcal{T}^2(\tilde{K})$.

A classic $\mathcal{B}$-valued restriction of this theorem is attributed to von Neumann (1944). In the present general $\mathcal{L}$-valued form, though operationnaly used in all bipolar-valued kernel computations from 1996 on, this result has not been thoroughly proved and published yet.

Based on this theorem, the following algorithm tackles the extension to the general case:

Algorithm 6 (Bisdorff 1997)**.**
Let $G^{\mathcal{L}}(X, \tilde{S})$ be a bipolar-valued outranking digraph.

1. Extract all outranking and outranked kernels $K_1, K_2, \ldots, K_j$ from the associated median cut graph $G(X, S)$.
2. Associate to each K_j a partially defined graph $G^{\mathcal{L}}_{K_j}(X, \tilde{S}_{/K_j})$ supporting exactly the unique kernel K_j.
3. Use the v. Neumann dual fixpoint iteration $\mathcal{T}^2$ for computing in turn $\tilde{K}_j$ in each partial graph $G^{\mathcal{L}}_{K_j}$.

A detailed description of this algorithm, with a partial proof of the correctness of the algorithm, may be found in Bisdorff [5].

A similar fixpoint based, but slightly more restricted, alogorithm for computing the $\mathcal{L}$-determined kernels may be deduced from the constructive proof of Theorem 2.

Algorithm 7 (Pirlot 2004)**.**
Let $G^{\mathcal{L}}(X, \tilde{S})$ be a bipolar-valued outranking digraph.

1. Extract all outranking and outranked kernels $K_1, K_2, \ldots, K_j$ from the associated median cut graph $G(X, S)$.
2. For each K_j: With $\tilde{Y}_0(x) = m$ for all $x \in K_j$ and $\tilde{Y}_0(x) = -m$ for all $x \notin K_j$, the iteration $\tilde{Y}_i = \mathcal{T}(\tilde{Y}_{i-1})$ for $i = 1, 2, \ldots$ converges to a fixpoint which is $\tilde{K}_j = \mathcal{T}(\tilde{K}_j)$.

3.4 Complexity

Except the first algorithm, which only applies to acyclic digraphs, both the Bisdorff and Pirlot algorithm rely on a first step which enumerates the kernels in the assocaited median cut crisp digraph.

For each crisp kernel solution, both fixpoint based algorithms compute the corresponding maximal sharp $\mathcal{L}$-valued result in at most $n \times |\mathcal{L}|$ steps, where each each step mainly involves two Boolean products of dimension $n \times 1$ and equality tests. Thus they operate in polynomial $\mathcal{O}(n \times |\mathcal{L}|)$ time, once the crisp kernels of the associated median cut digraph are available.

Main complexity remains thus definitely in the first step, i.e. enumerating all kernels in a crisp digraph.

References

[1] Berge, C., *The theory of graphs*. Dover Publications Inc. 2001. First published in English by Methuen & Co Ltd., London 1962. Translated from a French edition by Dunod, Paris 1958.

[2] Berge, C., *Graphes et hypergraphes*. Dunod, Paris, 1970

[3] Bisdorff, R. and Roubens, M., On defining fuzzy kernels from L-valued simple graphs. In: *Proceedings Information Processing and Management of Uncertainty, IPMU'96*, Granada, July 1996, 593–599.

[4] Bisdorff, R. and Roubens, M., On defining and computing fuzzy kernels from L-valued simple graphs. In: Da Ruan et al., eds., *Intelligent Systems and Soft Computing for Nuclear Science and Industry, FLINS'96 workshop*. World Scientific Publishers, Singapore, 1996, 113–123.

[5] Bisdorff, R., On computing kernels from L-valued simple graphs. In: *Proceedings 5th European Congress on Intelligent Techniques and Soft Computing EUFIT'97*, (Aachen, September 1997),vol. **1**, 97–103.

[6] Bisdorff, R., Logical foundation of fuzzy preferential systems with application to the Electre decision aid methods, *Computers & Operations Research*, **27**, 2000, 673–687.

[7] Bisdorff, R., Logical Foundation of Multicriteria Preference Aggregation. Essay in *Aiding Decisions with Multiple Criteria*, D. Bouyssou et al. (editors). Kluwer Academic Publishers, 2002, 379-403.

[8] Bisdorff, R., Pirlot, M., and Roubens, M., Choices and kernels in bipolar valued digraphs. *European Journal of Operational Research*, to appear.

[9] Bisdorff, Meyer P., and Roubens, M., Ruby, *Foundation of the RuBy methodology for solving the best choice problematics*. bf SMA working paper, University of Luxembourg, 2005. http://sma.uni.lu/bisdorff/Hyperkernels.pdf

[10] Codognet, P. and Diaz, D., Compiling Constraint in clp(FD). *Journal of Logic Programming*, Vol. 27, No. 3, June 1996.

[11] Diaz, D. *GNU-Prolog: A native Prolog Compiler with Constraint Solving over Finite Domains*. Edition 1.6 for GNU-Prolog version 1.2.13, 2002, http://gnu-prolog.inria.fr.

[12] Fernandez De la Vega, F., Kernels in random graphs. *Discrete Math.* 82 (1990), 213–217.

[13] Ghoshal, J., Laskar, R. and Pillone, D., Topics on domination in directed graphs. In Haynes, T.W., Hedetniemi, St. T., and Slater, P.J., Domination in graphs: Advanced topics. Marcel Dekker Inc. New-York – Basel, 1998, 401 – 437

[14] Haynes, T.W., Hedetniemi, St. T., and Slater, P.J., *Fundamentals of domination in graphs*. Marcel Dekker Inc., New-York – Basel, 1998.

[15] König, D., *Theorie de Endlichen und Unendlichen Graphen*. Chelsea, New York, 1950.

[16] Richardson, M. Solutions of irreflexive relations. *Ann. Math.* 58 (1953), 573 – 580.

[17] Roy, B. and Bouyssou, D., *Aide multicritère à la décision: Méhodes et cas*. Economica, Paris, 1993.

[18] Schmidt, G. and Ströhlein, Th., On kernels of graphs and solutions of games: a synopsis based on relations and fixpoints. *SIAM, J. Algebraic Discrete Methods*, **6**, 1985, 54–65.

[19] Schmidt, G. and Ströhlein, Th., *Relationen und Graphen*. Springer-Verlag, Berlin, 1989.

[20] Tomescu, I., Almost all digraphs have a kernel, *Discrete Math.*, 2, 84 (1990), 181–192.

[21] von Neumann, J. and Morgenstern, O., *Theory of games and economic behaviour*. Princeton University Press, Princeton 1944.