CHOOSING AMONG UNCERTAINTY MANAGEMENT TECHNIQUES: A CASE STUDY

ALESSANDRO SAFFIOTTI† AND ELISABETH UMKEHRER‡

IRIDIA - Université Libre de Bruxelles
Av. F. Roosvelt, 50 - CP 194/6
1050 Bruxelles - Belgium
E-mail: r01507@bbrbfu01.bitnet

Abstract

We present an experimental approach to comparing uncertainty management techniques. The need for comparison arises from the observation that each of the many existing techniques may be adequate for a different type of problems. Our goal is to find the technique that "best" fits a given problem. We analyze two problems, one of academic interest, and one tailored on a real application. We formalize and solve both problems by probabilities, belief functions, possibilities, and Boolean values. Based on this experience, we discuss some intrinsic properties of the considered theories, as well as the adequacy of each theory to model the given problem. The experiment is performed by using PULCinella, a general tool for Propagating Uncertainty based on the Local Computation technique of Shafer and Shenoy. Pulcinella may be specialized to each of the four uncertainty theories above. Moreover, Pulcinella allows the user to define easily his own specializations.

---

† Partially supported by the ARCHON project, which is funded by grants from the Commission of the European Communities under the ESPRIT-II Program, P-2256.

‡ Supported by a grant of IRIDIA, Université Libre de Bruxelles.

1. Introduction

The need to represent and reason with uncertain information in Knowledge-Based systems has been generally recognized in AI. To cope with this need, researchers in this field have developed a number of formal theories and techniques for managing uncertainty (see Saffiotti, 1987, for a review). While this proliferation of techniques has lead some authors to try to identify one "best" theory of uncertainty (e.g. Cheeseman, 1985), it seems to us that a more promising perspective from the AI viewpoint is to see uncertainty theories as alternatives, among which we can choose the one that best fits our needs (Szolovits and Pauker, 1978; J. Fox, 1986; Clark, 1988). Our aim would then be to find the most adequate technique for the uncertainty type affecting our problem. In order to do this, the first step is to formalize each uncertainty technique in a common framework, and to express the problem at hands in this framework. This step allows us to see how easily can our knowledge be expressed in each of the candidate techniques. As a possible outcome of this step, we may expect to discover that technique X requires data that is not available, or that technique Y cannot accommodate the data that is available. The second step is to compare the results obtained for our problem using the different techniques Differences in the results may originate from different ways data has been coded in each technique, or from differences in the inference mechanism themselves.

In this paper, we propose using PULCinella, a general system for Propagating Uncertainty based on Local Computation, as a tool for testing and comparing uncertainty management techniques. Pulcinella is an implementation of the general framework for local computation proposed by Shafer and Shenoy (1988), and it is fully described in (Saffiotti and Umkehrer, 1991a). Accordingly, it may be instantiated to any of the theories which have been formalized in this framework. In particular, four specialization of Pulcinella have already been implemented, namely for propagating probabilities, belief functions, Boolean values, and possibilities. Moreover, Pulcinella makes it easy to implement new theories in it (provided that they can be modelled in Shafer and Shenoy's formalism). The key for Pulcinella's comparison power lies in the separation made between the process of modelling the structural knowledge of a problem, and that of modelling its qualitative knowledge. Once a structural model has been decided, we can superimpose any of the available uncertainty calculi on it. We will illustrate our comparison procedure using Pulcinella by analyzing two cases. In each case, the same problem will be formalized in all the four specializations discussed above. The first case deals with a problem of academic interest: here, we will mainly concentrate on intrinsic differences between theories. The second case has been tailored on a real problem of diagnosing faults in electricity networks. Here, the knowledge engineer will find a discussion of the pros and cons of using different uncertainty theories for modelling the same test-bed problem.

The rest of this paper is organized as follows. Section 2 recalls some basic background on local computation, and presents the Pulcinella system. Section 3 and Section 4 present and discuss our two case studies. Section 5 concludes.

2. Pulcinella (Propagating Uncertainty by Local Computation)

2.1. A General Framework for Local Computation

Building on their work on belief function propagation (Shafer et al, 1987), Shenoy and Shafer (1988) propose a general framework for local computation. In this framework, the process of network propagation in itself has been abstracted from what is actually propagated. This framework has been further generalized by Shenoy (1989), who proposes a class of languages ("valuation-based languages") for building knowledge-based systems. These languages comprise objects, which are used to represent knowledge, and operators, which operate on these objects to make inferences on the knowledge. Two kinds of objects are considered, *variables* and *valuations*, and two operators, *combination* and *marginalization*[1]. We first remind the formal definitions of these elements, and will discuss their interpretation and use later.

Variables, Frames and Configurations. We consider a finite set of variables. Each variable may range over a finite set of possible values, called the *frame* for that variable. A *configuration* of a finite non-empty set of variables is an element of the cartesian product of the frames of the variables in this set.

Denotations: X for the complete set of variables; g, h, k for subsets of X;

W_g for the set of configurations of g; x,y for single configurations;

a,b,c for sets of configurations.

---

[1] A third operator, *solution*, is used for "decoding" the result obtained the propagation. This operator is not relevant to the present discussion, and so it will not be considered.

Sometimes we need to project a configuration of one set to another set. A configuration x of g is *projected* to h, $g \supset h$, by dropping all the elements in x belonging to g-h. It is *extended* to k, $k \supset g$, by building the cartesian product between the configuration and W_{k-g}.

Denotations: $x^{\downarrow h}$ for the projection of x to h; $x^{\uparrow h}$ for the extension of x to h.

Valuations. Given a set of variables h, we consider a set V_h. The elements of V_h are called *valuations* on the set h[2]. In our case, valuations are the objects that represent the uncertainty about a set of variables.

Denotations: V_g for the set of valuations on g; V for the set of all valuations on subsets of *X*; G, H for valuations.

Combination is any mapping $\otimes$: $V \times V \rightarrow V$, such that, if G and H are valuations on g and h, respectively, then $G \otimes H$ is a valuation on $g \cup h$.

Marginalization. For each $h \subseteq X$, there is a mapping $\downarrow h$: $\cup\{V_g \mid h \subseteq g\} \rightarrow V_h$, called *marginalization* to h, such that, if G is a valuation on g and $h \subseteq g$, then $G^{\downarrow h}$ is a valuation on h.

A set of variables, with their frames, together with a set of valuations, is called a *valuation system*. Once we have a valuation system, wa can evaluate it. This means to compute a global valuation on *X* obtained by combining together all the valuations in our valuation system, and find the marginals of this global valuation to each variable in *X*. Computing explicitly the global valuation is often unfeasible from the computational viewpoint. However, Shafer and Shenoy (1988) proposed a general local computation schema for evaluating valuation systems. The computation is local in the sense that combinations of valuations can be performed without extending each valuation to the whole space of the configurations. Shenoy and Shafer have shown that this schema can be applied if the combination and marginalization operators satisfy the following three axioms:

Axiom A1 (Commutativity and associativity of combination): Suppose G,H,K are valuations on g,h and k, respectively. Then

$$G \otimes H = H \otimes G \text{ and } G \otimes (H \otimes K) = (G \otimes H) \otimes K$$

---

[2] For the sake of simplicity we will not take into account in this paper "proper valuations", a subset of the valuations used to restrict the applicability the operators. Therefore, the definitions given here are not complete, but they preserve the basic ideas underlying valuation-based languages.

Axiom A2 (Consonance of marginalization): Suppose G is a valuation on g, and suppose $k \subseteq h \subseteq g$. Then

$$(G^{\downarrow h})^{\downarrow k} = G^{\downarrow k}.$$

Axiom A3 (Distributivity of marginalization over combination): Suppose G and H are valuations on g and h, respectively. Then

$$(G \otimes H)^{\downarrow g} = G \otimes (H^{\downarrow g \cap h})$$

2.2 Local Computation and Uncertainty Management

It will be useful to give now some examples of possible interpretations for the syntactical entities of a valuation-based language. This will show in which way a valuation-based language can be used for modelling different existing uncertainty theories. For probability theory, belief-functions, and a Boolean case, the mapping of the theory into the concepts of a valuation-based language have been proposed by Shenoy and Shafer (Shenoy and Shafer, 1988; Shenoy, 1989). For possibility theory, we use the mapping proposed by Dubois and Prade (Dubois and Prade, 1990). All these four interpretations satisfy axioms A1, A2 and A3 above. Thus, local computation can be used for them.

<u>Probability</u>:

Valuations on h are (unnormalized) probability distributions on the configurations of h

Combination: If G and H are probability distributions on g and h, respectively, then their combination is the probability distribution on $g \cup h$ defined by

$$(G \otimes H)(x) = G(x^{\downarrow g})H(x^{\downarrow h}) \quad \text{for all } x \in W_{g \cup h}.$$

Marginalization: If $h \subseteq g$ and G is a probability distribution on g, then the marginal of G for h is the probability distribution on h defined by[3]:

$$G^{\downarrow h}(x) = \Sigma\{G(x,y) \mid y \in W_{g-h}\} \quad \text{for all } x \in W_h$$

<u>Belief Functions:</u>

Valuations on h are basic probability assignment functions on sets of configurations of h.

---

[3] If g=h then $G^{\downarrow h}(x) = G(x)$. This applies also to the other interpretations given here.

Combination: If G and H are basic probability assignments on g and h, respectively, then their combination is the basic probability assignment on $g \cup h$ defined by[4]

$$(G \otimes H)(c) = \Sigma\{G(a)H(b) \mid (a^{\uparrow(g \cup h)}) \cap (b^{\uparrow(g \cup h)}) = c\} \text{ for all } c \subseteq W_{g \cup h},\ a \subseteq W_g, b \subseteq W_h$$

Marginalization: If $h \subseteq g$ and G is a basic probability assignment on g, then the marginal of G for h is the basic probability assignment on h defined by

$$G^{\downarrow h}(a) = \Sigma\{G(b) \mid b \subseteq W_g \text{ such that } b^{\downarrow h} = a\} \text{ for all } a \subseteq W_h.$$

Boolean:

Valuations on h are functions $H: W_h \rightarrow \{\text{true}, \text{false}\}$.

Combination: If G and H are valuations on g and h, respectively, then their combination is the valuation on $g \cup h$ defined, for all $x \in W_{g \cup h}$, by

$$(G \otimes H)(x) = \begin{cases} \text{true} & \text{if } G(x^{\downarrow g}) = \text{true and } H(x^{\downarrow h}) = \text{true} \\ \text{false} & \text{otherwise} \end{cases}$$

Marginalization: If $h \subseteq g$ and G is a valuation on g, then the marginal of G for h is the valuation on h defined, for all $x \in W_h$, by

$$G^{\downarrow h}(x) = \begin{cases} \text{true} & \text{if there is a } y \in W_{g-h} \text{ such that } G(x,y) = \text{true} \\ \text{false} & \text{otherwise} \end{cases}$$

Possibility:

Valuations on h are possibility distributions on sets of configurations of h.

Combination: If G and H are possibility distributions on g and h, respectively, then their combination is the possibility distribution on $g \cup h$ defined by

$$(G \otimes H)(x) = \min(G(x^{\downarrow g}), H(x^{\downarrow h})) \text{ for all } x \in W_{g \cup h}.$$

Marginalization: If $h \subseteq g$ and G is a possibility distribution on g, then the marginal of G for h is the possibility distribution on h defined by

$$G^{\downarrow h}(x) = \sup\{G(x,y) \mid y \in W_{g-h}\} \text{ for all } x \in W_h.$$

---

[4] This corresponds to usual Dempster's rule of combination (Dempster, 1966).

2.3. System's Overview

Pulcinella is a system for building and evaluating valuation systems based on Shenoy and Shafer's local computation technique. The system implements the general framework discussed above: it may be specialized to a given uncertainty theory by choosing an interpretation for the objects and the operators. Pulcinella is written in Lisp, and appears as a library of Lisp functions for creating, modifying and evaluating a valuation system. Alternatively, the user can choose to interact with Pulcinella via a graphical interface. The system builds on Hong Xu's implementation of the belief functions propagation technique of Shafer and Shenoy (Xu, 1991). The key move for generalizing Xu's program has been to parametrize it over a set of functions. At the moment, four interpretations of the general framework have been implemented: belief function, probability, Boolean and possibility (in the following, we will use the term "specialization" to refer to an implemented interpretation). By selecting one of these specializations, the user can transform Pulcinella in a probability propagation system, in a belief function propagation system, and so on. Moreover, Pulcinella is meant to be an open system: the set of functions which have to be defined to create a new specialization is very small and with a well defined semantics (reflecting the elements of a valuation-based language). Thus, it is easy for the user to define further specializations. We will first show how to use the existing specializations, and then discuss how a new specialization can be created.

As far as modelling quantitative knowledge is not concerned, the way to work with Pulcinella is the same for all specializations. The first thing the user has to do when he is face-to-face with Pulcinella, is to tell it which uncertainty theory he wants to use. Then, he can start to model his problem, either graphically or by calling Lisp functions. In terms of Shafer and Shenoy's framework, this means to create a valuation system representing his problem. This modelling process comprises two steps. In the first step, the user specifies the structural knowledge of his problem. This means defining all the variables to be used (along with their frames), and indicating which variables are linked together by a relation. By defining the variables and the relations the user implicitly fixes the subsets of variables for which valuations can be specified in the second step. The second step consists in modelling the quantitative knowledge: i.e. in defining the valuations on (some of) the subsets identified by the structural model created. This step depends from the specialization chosen. However, during this step, the user should keep in mind that a default valuation is given by the system to each variable and relation if no valuation is defined by the user. Once the valuation system has been completely defined, the user can evaluate it by asking Pulcinella to start propagation.

Pulcinella is an open system. The user can build his own specialization with Pulcinella. To build a new specialization the first thing he has to do is to express his theory in terms of the syntactical entities of valuation-based languages, and to prove that the axioms of local computation are satisfied. Having expressed the theory in this way, he may now implement the functions specific to the new specialization. This is made easier by the clear semantics given to these functions: basically, they mirror the concepts and entities of a valuation-based language. These functions may be divided up into three groups:

1st GROUP	2nd GROUP	3rd GROUP
init-value	combine	normalize
init-joint-value	marginalize	propagate-end
init-configuration		

The functions of the first two groups correspond to the syntactical entities of the valuation-based language. The first group collects the functions for initializing the objects of the language: "init-value" must specify the default valuation of a single variable; "init-joint-value" specifies the default valuation of a set of variables; and "init-configuration" builds the set of configurations corresponding to a set of variables. The last function is not necessary from a theoretical point of view, but it allows the user to choose different data-structures for representing configurations in different specializations (e.g. choosing to privilege efficiency vs. simplicity). These data-structures may be chosen for each specialization independently, but have to obey some restrictions[5].

The second group comprises the two functions that embody the combination and marginalization operators. This is clearly the conceptual kernel of a specialization.

The third group includes extra functions for changing the valuations in "some way". One function must implement the normalization procedure for valuations. The second function is called after the propagation has been completed, and allows the builder of a specialization to perform some housekeeping before the results are shown to the user.

---

[5] The suspicious reader may be reassured: these restrictions will not force him to use terribly complicated data-structures.

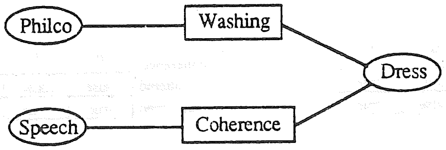
3. Case Study 1

3.1. The Problem

Our first example is adapted from (Saffiotti, 1987). We want to guess if Francesco will came wearing a black (B), white (W) or polka-dot (P) suite. We do not have any information about Francesco's preference (*state 0*), but we do know that his "Philco" washing machine is out; this makes us believe (say 80%) that he cannot choose W (*state 1*). Later, we remember that Francesco said yesterday that he dislikes mono-chromatic clothes; this is, for the notorious coherence of Francesco, a strong (say 90%) evidence both against B and W (*state 2*).

3.2. The Solutions

We first build a structural model for our problem. We define three variables: *Dress*, with frame {B, W, P}; *Philco*, with frame {ok, out}; and *Speech*, with frame {uttered, unuttered}. We then define two relations: *Washing*, between *Philco* and *Dress*; and *Coherence*, between *Speech* and *Dress*. The intended meaning of these elements should be self evident. The following is a graphical representation of our model, as appearing on the screen of Pulcinella:



The next step consists in deciding one uncertainty calculus to use, and to specialize Pulcinella to it. Suppose we choose to specialize Pulcinella to probability[6]. A dependency between variables is then encoded by a (unnormalized) joint probability distribution. The default valuation is the uniform probability distribution: if nothing is known about a variable or relation, all configurations are considered to have the same probability. We can then enter the distributions for our variables and relations. These distributions, which encode the quantitative knowledge in our problem, are shown below[7]:

---

[6] In practice, this reduces to selecting a menu item, or to evaluating the form "(specialize-uncertainty 'probability)".

[7] The valuations for these variables are obvious, and will not considered again in the next examples.

possible, and is expressed by a mapping from configurations to the interval [0,1]. The default possibility distributions is given by attaching to each configuration the value 1: if nothing is known about a set of variables, all configurations are regarded as completely possible. After switching Pulcinella to the possibilistic interpretation, we enter two possibility distributions for our relations. These distributions, and the one obtained for the variable *Dress* after propagation, are shown below.

$\Pi_{washing}$	B	W	P
ok	1	1	1
out	1	0.2	1

$\Pi_{coherence}$	B	W	P
uttered	0.1	0.1	1
unuttered	1	1	1

Value	*State 0*	*State 1*	*State 2*
B	1	1	0.1
W	1	0.2	0.1
P	1	1	1

Finally, we consider the case in which we collapse uncertainty to true/false values. These values are better understood in term of satisfaction of constraints: a value true for a configuration means that this configuration is acceptable given the constraints that characterize our problem. Accordingly, a relation among variables will be encoded by selecting all those configurations that are admissible, and by associating true to them. The default valuation is given by attaching to each configuration the value true: if nothing is known about a set of variables, each configuration can be true. The following tables show the values for our relations, and the results obtained, with the Boolean specialization.

$\mathcal{T}_{washing}$	B	W	P
ok	true	true	true
out	true	false	true

$\mathcal{T}_{coherence}$	B	W	P
uttered	false	false	true
unuttered	true	true	true

Value	*State 0*	*State 1*	*State 2*
B	true	true	false
W	true	false	false
P	true	true	true

3.3. Discussion

In this example, as in the next one, we can recognize three basic categories of differences:

1) differences in the data we have to provide (which data, how many, in which form, . . .);

2) differences in the results due to differences in the data we provided; and

$\mathcal{P}_{washing}$	B	W	P
ok	1/3	1/3	1/3
out	0.4	0.2	0.4

$\mathcal{P}_{coherence}$	B	W	P
uttered	0.05	0.05	0.9
unuttered	1/3	1/3	1/3

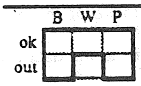
$\mathcal{P}_{Philco}$	
ok	0
out	1

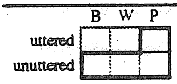
$\mathcal{P}_{Speech}$	
uttered	1
unuttered	0

Finally, we ask Pulcinella to start propagation. The following table gives the marginal probabilities computed for *Dress* at different moments[8]:

Value	*State 0*	*State 1*	*State 2*
B	0.33	0.4	0.05
W	0.33	0.2	0.03
P	0.33	0.4	0.92

Suppose that we now want to try to use belief functions for modelling our problem. All we have to do, is to specialize Pulcinella to belief functions, and to input the quantitative knowledge of the problem in the form of two basic probability assignments representing our relations:

Subset	$m_{washing}$
	0.8

Subset	$m_{coherence}$
	0.9

Intuitively, the subset to which $m_{washing}$ assigns a 0.8 mass represents the fact that the answer to our problem may be any of B, W and P when *Philco* = <u>ok</u>, and any of B and P when *Philco* = <u>out</u>. The remaining 0.2 mass is automatically given to the whole frame. The default valuation is the basic probability assignment function which attach the value 1 to the whole set of configurations. After propagation, we get the following results for the variable *Dress*[9]

	State 0		*State 1*		*State 2*	
Value	bel	pl	bel	pl	bel	pl
B	0	1	0	1	0	0.1
W	0	1	0	0.2	0	0.02
P	0	1	0	1	0.9	1

Next, we consider using possibility theory. In this specialization, the user models quantitative knowledge by specifying possibility distributions on the defined sets of variables. A possibility distribution on a set of variables reflects to what extend each configuration of the set of variables is

---

[8] States in the table refer to the states mentioned in the statement of our story.

[9] For greater readibility, we show the results using *bel* and *pl* functions (Shafer, 1976).

3) differences in the results due to differences in the mechanisms used in the theories.

Concerning category 1 above, the quantitative knowledge stated in the problem has to be coded differently in the four specializations. In the probabilistic case, the need to specify completely the joint distributions $\mathcal{P}_{\text{washing}}$ and $\mathcal{P}_{\text{coherent}}$ has obliged us to replace in some way the missing information. In particular, in $\mathcal{P}_{\text{washing}}$, the 80% probability of $(B \vee W)$ has been converted into exact probability values for B and P individually (arbitrary assuming equiprobability); a similar operation has been made for $\mathcal{P}_{\text{coherent}}$. On the other hand, in both the belief function and the possibilistic specializations knowledge has been expressed at exactly the level of granularity that is available. It must be noticed that using basic probability assignments to encode knowledge in the belief function specialization may sometimes be less intuitive (mainly because we have to work on subsets). Finally, our knowledge has easily been reduced in the obvious way in the Boolean specialization.

Moving now to the analysis of the results (categories 2 and 3 above), we first notice that —as expected— all four specializations agree from the qualitative viewpoint. The major quantitative differences show up between the results of the probabilistic specialization and those of the belief function (or possibilistic) one. These differences must be tracked back to differences in the data used, mainly because of the additional hypotheses (equiprobability) introduced in the probabilistic case. Remarkably, these differences are particularly evident in those cases (States 0 and 1) in which ignorance is predominant: here, in spite of our ignorance, the probabilistic approach needs (and gives) precise figures[10]. Another interesting difference arises between the belief function and the possibilistic cases. Here, differences in results do not originate from differences in the inputs given, but rather from the different combination mechanisms. To wit, consider State 2: the Philco evidence and the Speech evidence are both denying the W hypothesis. These two items of evidence have been combined in one much stronger evidence against W by Dempster's rule in the belief function case, resulting in a very small (0.02) plausibility for W. On the other hand, the stronger of them has simply been selected through the MIN operator in the possibilistic approach, giving a possibility value of 0.1 for W.

---

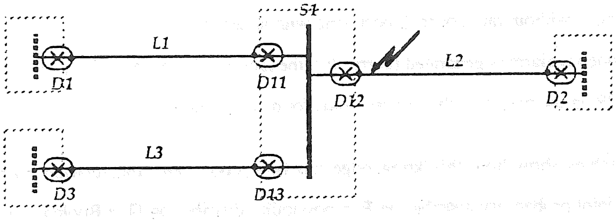
[10] «Wovon man nicht sprechen kann, darüber muß man schweigen» (What we cannot speak about we must consign to silence) (Wittgenstein, 1921).

4 Case Study 2

The above considerations illustrates how to use Pulcinella for analyzing the different behaviours of uncertainty treating theories. However, we do not have up to now a pragmatic unit for measuring how much do the different theories fit our needs. Still, Pulcinella may be also used as a tool for evaluating uncertainty formalisms in the light of the uses to which they are to be put. To this respect, we now present our second case.

4.1. The Problem

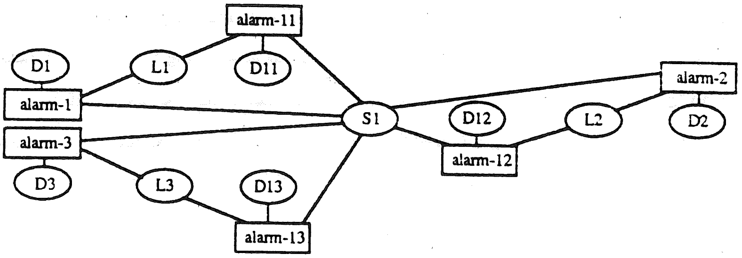
This problem has been tailored on an experiment made in modelling the uncertainty present in a problem of fault diagnoses in electricity networks. For the sake of clarity, both the qualitative and the quantitative knowledge have been greatly simplified. The full experiment, and the actual figures used, are reported in (Saffiotti and Umkehrer, 1991b). We consider here the following fragment of an electricity network:



This fragment comprises four substations, linked by three lines L1, L2 and L3. The substation in the middle includes S1, a big conductive bar used for connecting more lines together. The Di's are "circuit breakers": devices that can separate two lines, and that watch the part of the network on their "hot" side (marked by a dot in the picture) for overloads. When an overload is detected, a circuit breaker may generate two kinds of alarms: "instantaneous", for "big" overloads (normally caused by a fault in the line the device is on); or "delayed", for "small" overloads (normally caused by a fault in a neighbour line).

4.2. The Solutions

We can model the above description in Pulcinella by the following variables and relations:



where *Di*'s are variables representing circuit breaker states, with possible values ok (no alarm), del (delayed alarm), and inst (instantaneous alarm); *Li*'s and *S1* represent line states, with frame {ok, fault}; and the *alarm-i*'s relate generation of alarms by breakers with states of neighbour lines.

The quantitative knowledge given by the experts is not very rich. Essentially, it says that:

1. alarms are not very reliable: in roughly 10% of the cases, they do not correspond to the real situation (alarm generated without fault, or fault occurring without alarm);
2. if an instantaneous alarm is generated (correctly), the fault is 70% of the cases in the line the breaker is on, and 30% in the next one; the reverse holds for delayed alarms.

The following tables show how this knowledge has been coded into the relation *alarm-1* with an (unnormalized) joint probability distribution $\mathcal{P}$, a possibility distribution Π, a Boolean constraint $\mathcal{T}$, or a basic probability assignment $\boldsymbol{m}$, respectively. Distributions for the other *alarm-i*'s are similar; however, those for the *alarm-ij*'s are different.

$\mathcal{P}(\cdot)$

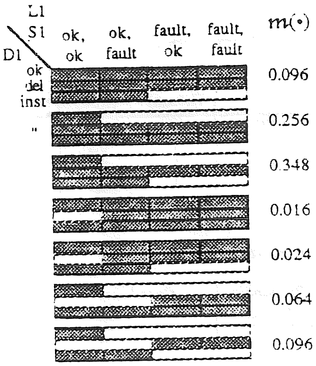
L1, S1	D1 ok	D1 del	D1 inst
ok, ok	0.9	0.1	0.1
ok, fault	0.05	0.6	0.2
fault, ok	0.05	0.2	0.6
fault, fault	0.001	0.1	0.1

$\Pi(\cdot)$

	ok	del	inst
ok, ok	1	0.1	0.1
ok, fault	0.1	0.7	0.3
fault, ok	0.1	0.3	0.7
fault, fault	0.1	1	1

$\mathcal{T}(\cdot)$

	ok	del	inst
ok, ok	true	false	false
ok, fault	false	true	true
fault, ok	false	true	true
fault, fault	false	true	true



We now imagine a scenario in which a fault occurs on line L2 very near to D12 (see the above picture): consequently, a "delayed" alarm is sent by D2, and a "instantaneous" alarm by D12. Moreover, we imagine that D1 has been adjusted incorrectly (it happens), and is too sensitive: thus, also D1 sends a "delayed" alarm. Three main hypotheses are compatible with this set of alarms, shown in order of preference:

1) Fault in L2; alarm from D1 spurious;
2) Fault in S1; missing alarm from D3, and alarm from D12 spurious;
3) Fault in L1; D11 not working (missing alarm & did not open), missing alarm from D3, and alarm from D12 spurious.

The following tables summarize the results obtained by running Pulcinella on this example with the probability, possibility, Boolean, and belief function specializations, respectively.

1. After receiving the alarm from D2

Var	*Probability*	*Possibility*		*Belief Functions*		*Boolean*	
	$\mathcal{P}$(fault)	Π(fault)	Π(ok)	bel(fault)	bel(ok)	$\mathcal{T}$(fault)	$\mathcal{T}$(ok)
L1	0.002	1	0.7	0.02	0	true	true
L2	0.226	1	0.7	0.02	0	true	true
L3	0.002	1	0.7	0.02	0	true	true
S1	0.087	1	0.3	0.16	0	true	true

2. After receiving the alarm from D12

	Probability	*Possibility*		*Belief Functions*		*Boolean*	
Var	$\mathcal{P}$(fault)	Π(fault)	Π(ok)	bel(fault)	bel(ok)	$\mathcal{T}$(fault)	$\mathcal{T}$(ok)
L1	0.002	1	0.7	0.02	0	true	true
L2	0.972	1	0.1	0.60	0	true	false
L3	0.002	1	0.7	0.02	0	true	true
S1	0.013	1	0.3	0.16	0	true	true

3. After receiving the alarm from D1

	Probability	*Possibility*		*Belief Functions*		*Boolean*	
Var	$\mathcal{P}$(fault)	Π(fault)	Π(ok)	bel(fault)	bel(ok)	$\mathcal{T}$(fault)	$\mathcal{T}$(ok)
L1	0.219	1	0.7	0.03	0	true	true
L2	0.919	1	0.1	0.60	0	true	false
L3	0.002	1	0.7	0.03	0	true	true
S1	0.141	1	0.3	0.29	0	true	true

4.3. Discussion

As in Section 3, we first consider the differences in the inputs required by the different techniques (category 1 above), and than the differences in the output produced (categories 2 and 3). However, we try to take here a more "knowledge engineering" viewpoint. Probability theory requires several values that are not available, or not directly available, in our data (e.g. $\mathcal{P}$({ok, ok, fault}) and $\mathcal{P}$({ok, fault, fault}): these values had to be computed from the available information, by introducing hypotheses of equiprobability or independence. Notice that the amount of subjective estimates required by the probabilistic approach greatly increases the risk of inconsistency among them (but see Duda et al., 1986). On the contrary, the process of supplying data is extremely simple in the possibility and the belief function specializations: the expert must supply just the data she knows, and consign the rest to silence. We do not need to force her (or the figures she supplies) to say something they were not meant to. However, in the belief function case, the translation of this data into basic probability assignments may sometimes be hard. The difficulty of the Boolean case is somewhat complementary to that of the probabilistic case: the data given by the expert has to be rounded very roughly, and she may feel uncomfortable with the approximations obtained. A possible reaction to this rude attitude is to try to split general rules into more specific ones. The aim of this would be to reduce uncertainty by explicitly accounting for the possible exceptions to rules. Though this constitutes a stimulus for the expert that may sometimes result in a better explication of her knowledge, the strongly empirical, tangled and "artistic" nature of electrical fault diagnostic knowledge discouraged us from using the Boolean specialization for our problem.

If we now switch to considering the results of our example, we can observe that our four specializations have produced results that are both quantitatively and qualitatively different. The first phenomenon we notice is that the Boolean specialization does not suggest the possibility of a fault in S1. The causes of this reside both in the input given and in the combination mechanism used. As for the input, the knowledge encoded in our relations allows us to infer that the fault is, e.g., either in S1 or in L1. Yet, no relation encodes knowledge which allows us to infer the S1 hypothesis alone. Discrimination between S1 and L1 is, on the other side, captured by differences in the given weights in the other approaches. As for the mechanism, aggregating knowledge by an AND operation does not allow to perform that "counting" of evidence that seems necessary if we want to accumulate items of weak evidence together. In our example, the evidence given by D2 only partly supports the hypothesis S1; however, we expect that further support to S1 coming from D1 should reinforce the S1 hypothesis. One way to fix this problem in the Boolean specialization is to add new rules (e.g. "IF at least 2 among D1, D2 and D3 send a delayed alarm, THEN S1 is faulty"); but the lesson taken from this story seems to be that considering uncertainty in our problem is necessary, if we want to preserve inferential power without having to drown in a see of specialized rules.

The second qualitative difference we want to notice regards the hypothesis "*L1* = fault", which is suggested by the possibilistic and the probabilistic[11] specializations, but not by the belief function one. The origin of this is again in the way we have coded our data: a delayed alarm does not support, in the belief function case, a fault in the adjacent line *individually*, but a fault in the adjacent *or* in the next lines[12]. On the contrary, in the possibilistic and the probabilistic cases, the evidence given by the alarm has to be spread among each hypothesis individually. A related difference concerns the hypothesis "*L3* = fault", which is suggested by the possibilistic specialization only. This "over-inferencing" is rooted in the fact that we have taken a rather strong attitude when defining the possibility distributions for the *alarm-i*'s relations (viz. the frequential knowledge about the localization of the fault has been converted

---

[11] Notice that the possibilistic specialization supports this hypothesis by partially denying that "*L1* = ok".

[12] But notice that, differently from the Boolean case, further evidence may convert this support to a support for the adjacent line individually.

to a measure of (im)possibility). A less committed interpretation of our data would lead to the following distribution

$\Pi(\bullet)$	ok	del	inst
ok, ok	1	0.1	0.1
ok, fault	0.1	1	1
fault, ok	0.1	1	1
fault, fault	0.1	1	1

where the information about the relative support given by an alarm to a near fault or a far fault (which is not, strictly speaking, a matter of possibility) has been ignored. Using this encoding, we would get a result suggesting L2 alone (like in the Boolean case). The lesson to be taken here seems to be that possibility theory may provide the additional inference power that we need in our problem, but this power cannot be "fine-tuned" easily.

As for the quantitative differences, the most important one (which might be regarded as qualitative as well) concerns the hypothesis "S1 = fault". This hypothesis is reinforced by the arrival of the alarm from D1 in the probabilistic and belief function specializations, but not in the possibilistic one. The cause here is only the combination mechanism used: like the AND of the Boolean case, the MIN operator does not allow us to perform that "counting", which seems necessary in our problem to accumulate evidence correctly. As a consequence, the answers given by the possibilistic specialization to our problem are mainly meaningful from the qualitative viewpoint (they allow us to focus on those hypotheses which are possible), but they are fairly poor from the quantitative one. To this respect, we notice that the results given by the probabilistic specialization are very rich from the quantitative viewpoint; though, because of our straining the input data, this precision may be somehow unjustified. The belief function specialization seems to provide the tradeoff between precision and non-commitment that best fits our knowledge. Yet, the computational complexity inherent in belief functions shows up in running the full experiment: the final choice of the uncertainty treating technique to be used for solving the full scale diagnostic problem will have to take this factor into consideration.

5. Conclusions

Whenever we have to deal with a problem involving uncertain knowledge, choosing the uncertainty management technique which best fits it becomes of primary importance. To this respect, the approach we have proposed is peculiar in its being experimentally grounded. There are at least two places where such an approach, and the tool it is based on, is expected to be useful. First, in the room of the theorist involved in the comparative study of uncertain reasoning models. For her, Pulcinella might play the role of a slide-rule in her testing the behaviour of uncertainty theories over sample problems of academic interest. The fact that the structural model of the problem is the same for all theories ensures the soundness of the test. Second, in the office of the knowledge engineer. Here, Pulcinella might prove helpful in checking out different uncertainty management techniques for solving the problem at hands, and then judging—on this experimental basis—which one appears to be the most adequate to our case. In particular, we may test both the input requirements and the obtained results of each theory, and evaluate how the available data is accommodated for, or our expectations satisfied.

Pulcinella belongs to the growing community of systems for representing and propagating uncertainty in networks (e.g. Andersen et al., 1989; Hsia and Shenoy, 1989; Zarley et al., 1988; Xu, 1991). However, to our knowledge, it is the only existing system that can propagate uncertainty values according to more than one uncertainty theory.

Acknowledgements

The present research has benefit from discussions with (and comments from) Fabrizio Sebastiani, Yen-Teh Hsia, Robert Kennes, Bruno Marchal, Philippe Smets and Xu Hong.

References

Andersen, S.K., Olesen, K.G., Jensen, F.V. and Jensen, F. (1989) "HUGIN - a Shell for Building Bayesian Belief Universes for Expert Systems", *Procs. of the 11th Int. Congress on AI* (Detroit).

Chatalic, P., Dubois, D. and Prade, H. (1987) "A System for Handling Relational Dependencies in Approximate Reasoning", *Procs. of the 3rd Int. Expert System Conf.* (London, GB) 495-502.

Cheeseman, P. (1985) "In Defense of Probability", *Procs. of IJCAI-85*: 1002-1009.

Clark, D. (1988) (Ed.) "Responses to 'An AI view of the treatment of uncertainty' by Alessandro Saffiotti", *The Knowledge engineering Review* 3(1) 59-91.

Dempster, A.P. (1966) "Upper and Lower Probabilities Induced by a Multivalued Mapping", *Annals of Mathematical Statistic* **38**: 325-339.

Dubois, D. and Prade, H. (1990) "Inference in possibilistic hypergraphs", Proc. of IPMU Conference.

Duda, R.O., Hart, P.E. and Nilsson, N.J. (1986) "Subjective Bayesian Methods for Rule-Based Inference Systems", *AFIPS Procs.* **45** (New York) 1075-1082.

Fox, J. (1986) "Three Arguments for Extending the Framework of Probability", in Kanal, L.N. and Lemmer, J.F. (Eds.) *Uncertainty in AI* (North Holland).

Hsia, Y. and Shenoy, P.P. (1989) "MacEvidence: A Visual Environment for Constructing and Evaluating Evidential Systems", *Working Paper No. 211* (School of Business, University of Kansas, Lawrence).

Lauritzen, S.L. and Spiegelhalter, D.J. (1988) "Local Computations with Probabilities on Graphical Structures and Their Applications to Expert Systems", *Journal of the Royal Statistical Society B* **50**(2) 157-224.

Pearl, J. (1988) *Probabilistic Reasoning in Intelligent Systems* (Morgan Kaufman, CA).

Saffiotti, A. (1987) "An AI view of the treatment of uncertainty", *The Knowledge Engineering Review* 2(2) (Cambridge University Press, UK) 75-97.

Saffiotti, A. (1989) "De Propagationibus", *ARCHON Technical Note IRIDIA/ARCHON/TN-05*, (IRIDIA, Université Libre de Bruxelles, Bruxelles).

Saffiotti, A. and Umkehrer, E. (1991a) "PULCINELLA User's Manual", *Technical Report IRIDIA/91-5*, (IRIDIA, Université Libre de Bruxelles, Bruxelles).

Saffiotti, A. and Umkehrer, E. (1991b) "What do you want to propagate for breakfast ?", *Technical Report IRIDIA/91-7*, (IRIDIA, Université Libre de Bruxelles, Bruxelles).

Shafer, G. (1976) *A Mathematical Theory of Evidence* (Princeton University Press, Princeton).

Shafer, G., Shenoy, P.P. and Mellouli, K. (1987) "Propagating Belief Functions in Qualitative Markov Trees", *International Journal of Approximate Reasoning* **1**: 349-400.

Shenoy, P.P. (1989) "A Valuation-based Language for Expert Systems", *International Journal of Approximate Reasoning* **3**: 383-411.

Shenoy, P.P. and Shafer, G. (1988) "An Axiomatic Framework for Bayesian and Belief-Function Propagation", Proc. of AAAI Workshop on Uncertainty in AI: 307-314.

Smets, Ph. (1988) "Belief Functions", in: Smets, Ph., Mamdani, E.H., Dubois, D. and Prade, H. (Eds.) *Non-Standard Logics for Automated Reasoning* (Academic Press, London).

Szolovits, P. and Pauker, S.G. (1978) "Categorical and Probabilistic Reasoning in Medical Diagnosis", *Artificial Intelligence* **11**: 115-144.

Wittgenstein, L. (1921) *Tractatus Logico-Philosophicus* (Annalen der Naturphilosophie).

Xu, H. (1991) "An Efficient Implementation of Belief Function Propagation", *Technical Report IRIDIA/91-4*, (IRIDIA, Université Libre de Bruxelles, Bruxelles).

Zarley, D.K., Hsia, Y. and Shafer, G. R. (1988) "Evidential Reasoning using DELIEF", *Working Paper No. 193* (School of Business, University of Kansas, Lawrence).