Extending Uncertainty Formalisms to Linear Constraints and Other Complex Formalisms [⋆]

Nic Wilson

Cork Constraint Computation Centre,
Department of Computer Science,
University College Cork, Cork, Ireland
n.wilson@4c.ucc.ie

Abstract

Linear constraints occur naturally in many reasoning problems and the information that they represent is often uncertain. There is a difficulty in applying AI uncertainty formalisms to this situation, as their representation of the underlying logic, either as a mutually exclusive and exhaustive set of possibilities, or with a propositional or a predicate logic, is inappropriate (or at least unhelpful). To overcome this difficulty, we express reasoning with linear constraints as a logic, and develop the formalisms based on this different underlying logic. We focus in particular on a possibilistic logic representation of uncertain linear constraints, a lattice-valued possibilistic logic, an assumption-based reasoning formalism and a Dempster-Shafer representation, proving some fundamental results for these extended systems. Our results on extending uncertainty formalisms also apply to a very general class of underlying monotonic logics.

Key words: possibilistic logic, lattice-valued possibilistic logic, Dempster-Shafer theory, assumption-based reasoning, linear constraints, spatial and temporal reasoning

1 Introduction

Many reasoning problems involve linear constraints restricting the possible values of real-valued variables; in particular temporal and spatial problems can involve linear constraints representing relationships between temporal variables and between spatial variables, see e.g., [16,26,7]. Such constraints can often

[⋆] This paper is an extended version of (Wilson, 2004) [34].

represent information that is uncertain. Many formalisms for representing and reasoning with uncertain information have been developed. The underlying logical information is typically expressed as a finite set of possibilities, or using propositional calculus, or sometimes first order predicate calculus. One can sometimes convert linear constraints to a discrete (e.g., propositional) form, but this can make the representation very cumbersome, and the important metric information will tend to be hidden. Furthermore, discrete representations cannot take the continuous nature of space and time into account, for which it can be natural to have continuously graded representations of uncertainty. For example, based on different sets of map data, we may have (for example, in a Dempster-Shafer representation) a degree of belief of 0.8 that a well is more than 5 metres from a property boundary, and a degree of belief of 0.7 that it is more than 8 metres away. The degree of belief that it is more than 7 metres from the boundary will tend to be more than 0.7 and less than 0.8; in fact, the degree of belief may well vary continuously with the distance from the boundary.

It can therefore be preferable and more natural to represent linear constraints (and more general kinds of constraint) directly, and extend the uncertainty theories to reason with these. We approach this problem by expressing linear constraints in a logic (Section 2.1) and generalising uncertainty formalisms by defining them over this logic. These are illustrated in terms of a simple example, based on a real application of reasoning with uncertain geographic information. Our approach to generalising uncertainty formalisms applies for a very general class of underlying logics, which we define formally in Section 2.2; this includes logics which can reason with disjunctions of linear constraints, or even non-linear constraints, allowing, for example, more expressive representations of spatial boundaries.

The basic idea behind the generalisation of uncertainty formalisms is to consider the uncertainty theories over (finite or infinite) sets of possibilities, and associate a constraint with its semantics. For example, suppose that c is a constraint on a set of real-valued variables, and let $[c]$ be the set of all assignments that satisfy the constraint c. Then (for example) the degree of belief of c is defined to be the degree of belief of the set $[c]$. This kind of approach can be used to extend logic-based formalisms, such as various non-monotonic logics and belief revision formalisms. In this paper we focus on formalisms which involve some kind of grade or degree of support being allocated to propositions. The grades of support may be totally ordered: as in Dempster-Shafer beliefs, or necessity values in possibility theory and possibilistic logic; or only partially ordered as in lattice-valued possibilistic logic and the generalised assumption-based reasoning formalism. Partially ordered degrees of support can be natural, for example, when one has information from two sources, and we don't know which of the sources is more reliable. The degrees of belief are interpreted probabilistically in Dempster-Shafer theory, and more qualitatively in the other formalisms considered, where the grades may just represent ordering information amongst strengths of belief.

We consider possibilistic logic in Section 3, where we show (Theorem 2) how deduction in the possibilistic logic can be achieved using deduction in the underlying logic. In Section 4 we generate a system of lattice-valued possibilistic logic, based on linear constraints or other underlying monotonic logics; in a similar way we generate a generalised assumption-based reasoning formalism. For each of these we give a sound and complete proof theory. In Section 5 we show how one can extend Dempster-Shafer theory.

2 Underlying Monotonic Logics

In this section we define the underlying logics for the generalised uncertainty formalisms constructed in Sections 3, 4 and 5. First we consider a logic of linear constraints. Then, in Section 2.2 we consider a general class of monotonic logics.

2.1 A Logic of Linear Constraints

We describe, in this section, a logical representation of linear $\geq$-constraints, with a semantics and a proof theory that is sound and complete for finite sets of constraints. We consider linear constraints of the following form: $a_1x_1+\cdots+a_nx_n \geq a_0$, where $a_0, ..., a_n$ are known real numbers, and $x_1, ..., x_n$ are unknown real numbers, often representing some physical quantities that we're interested in, but only have partial information about. This constraint is saying that the unknown vector $x = (x_1, \ldots, x_n)$ must be such that $a_1x_1 + \cdots + a_nx_n \geq a_0$ holds.

The language. Let $V = \{X_1, \ldots, X_n\}$ be a finite set of real-valued variables.[1] We are interested in linear constraints on V of the form $a_1X_1 + \cdots + a_nX_n \geq a_0$. Formally we define a (linear) constraint a to be a real-valued function on $\{0, \ldots, n\}$, where $a(i)$ is usually written a_i. Let $\mathcal{L}$ be the set of all such (linear) constraints. Define a model x to be a real-valued function on the set $\{1, \ldots, n\}$. $x(i)$, usually written x_i, is interpreted as a value of the variable

[1] Each variable is assumed to have a true, but (usually) unknown, value. We do not consider here the more complex case where some of the variables are decision variables, as studied in e.g., Simple Temporal Problems under Uncertainty (Vidal and Fargier, 1999) [30].

X_i. Let $\mathcal{M}$ be the set of all models. We say that model x satisfies a, written $x \models a$, if and only if $a_1x_1 + \cdots + a_nx_n \geq a_0$.

Element a of $\mathcal{L}$ represents the inequality: $\sum_{i=1}^n a_iX_i \geq a_0$. We will usually slightly abuse the notation and write elements of $\mathcal{L}$ as linear inequalities $\sum_{i=1}^n a_ix_i \geq a_0$, or some equivalent form such as $\sum_i -a_ix_i \leq -a_0$.

We label three special constraints as $\top$, $\top^0$ and $\bot$, which are defined as follows: for each $i \in \{1, \ldots, n\}$, $\top(i) = \top^0(i) = \bot(i) = 0$, and $\top(0) = -1$, $\top^0(0) = 0$ and $\bot(0) = 1$. Thus $\top$ can be considered as the constraint $0 \geq -1$, $\top^0$ as $0 \geq 0$ and $\bot$ as $0 \geq 1$. $\top$ and $\top^0$ are satisfied by every model x, and $\bot$ is satisfied by none. Constraints can be added, and multiplied by real valued scalars: for constraints a, b and real number r, constraint $a + b$ is defined by $(a + b)_i = a_i + b_i$ for all i, and ra is defined by $(ra)_i = ra_i$ for all i.

The language can also be used to represent constraints with $\leq$ replacing $\geq$, and also linear equalities. A constraint $a_1x_1 + \cdots + a_nx_n \leq a_0$ can be written as $(-a_1)x_1 + \cdots + (-a_n)x_n \geq -a_0$, so is equivalent to the constraint $-a$. (x satisfies the former constraint if and only if it satisfies $-a$.) The linear equality $a_1x_1 + \cdots + a_nx_n = a_0$ holds if and only if both $a_1x_1 + \cdots + a_nx_n \geq a_0$ and $a_1x_1 + \cdots + a_nx_n \leq a_0$ hold so is equivalent to the pair of constraints $\{a, -a\}$. We could also easily extend the language to include strict constraints of the form $a_1x_1 + \cdots + a_nx_n > a_0$ though, to keep the language simpler, we do not do so here.

Here are a few examples of constraints that can be represented using this language: (1) $x_1 \geq 5.2$; (2) $x_1 \leq x_2$; (3) $x_2 = x_1 - x_3$; (4) $x_1 - 2x_2 \leq x_3 + 3x_4$; (5) $x_1/(x_1 + x_2 + x_3) \in [0.5, 0.6]$; (6) the absolute value of $x_1 - x_2$ is not more than 5; (7) the arithmetic mean of x_1, x_2 and x_3 is 4.3.

Consistency and semantic consequence. Suppose we have a set A of constraints on unknown x. We say, in the usual way, that x satisfies A (written $x \models A$) if and only if x satisfies every member of A, i.e., $x \models a$ for all $a \in A$. Let $[A]$ be the set of $x \in \mathcal{M}$ that satisfy A, i.e., $[A] = \{x \in \mathcal{M} : x \models A\}$. A is said to be consistent if it has a model, i.e., if $[A]$ is non-empty; otherwise it is said to be inconsistent. We would like to be able to talk about what constraints b necessarily follow from those in A. Formally we define semantic consequence relation $\models$ by $A \models B$ *if and only if every element b of B is satisfied by every model of A, i.e.,* $[A] \subseteq [B]$. Set of constraints A is inconsistent if and only if $A \models \{\bot\}$, since $\bot$ has no model. By its construction, semantic consequence $\models$ is a reflexive, transitive and hence monotonic consequence relation. However, it is not compact; for example, if a^k is the constraint $x_1 \geq k$ then $A = \{a^k : k = 1, 2 \ldots\}$ is inconsistent, but every finite subset of A is consistent.

Syntactic consequence. Consider the proof theory generated by the axioms $\top$ and $\top^0$ and inference rule schemas (where a and b are arbitrary elements of $\mathcal{L}$):

For any real $r > 0$, *From* a *deduce* ra.

From a *and* b *deduce* $a + b$.

For any constraint a, *From* $\perp$ *deduce* a.

For set of constraints A and constraint b we say in the usual way that b can be proved from A, written $A \vdash b$, if b can be derived from applying iteratively the above inference rules to A and the axioms $\top$ and $\top^0$; define also $A \vdash B$ if $A \vdash b$ for all $b \in B$.

Any such (finitary) syntactic consequence relation $\vdash$ is compact by definition, so we can't hope for full completeness, as $\models$ is not compact. However, we have, by well-known fundamental results for linear programming (see e.g., Chapter 1 of (Stoer and Witzgall, 1970) [29]) the following result (see (Wilson, 2002) [33]).

Theorem 1 *[Finite Completeness] For any sets of constraints A and B, $A \vdash B$ implies $A \models B$. If furthermore, A is finite then $A \vdash B \iff A \models B$.*

In practice, one will use more developed tools for finding the consequences of a set of such constraints: for general problems, linear programming techniques; for particular sparse systems, Fourier elimination can be efficient; or fast algorithms for special kinds of constraints, such as Simple Temporal Networks (Dechter et al, 1991) [7].

The expression of reasoning with linear constraints as a logic makes it easy to generalise many (in particular non-monotonic) extensions of classical logics to linear constraints. The logic described above is closely related to the logic of probability described in (Wilson and Moral, 94) [36] the main difference being that the latter has some additional axioms, because of models being probability functions which are non-negative. The methods for producing non-monotonic extensions to this logic of probability can be adapted to produce non-monotonic logics of linear constraints. In particular, the definition of a default logic of probability in [36] (related to Reiter's default logic [25]) carries over immediately to a default logic of (finite sets of) linear constraints; this involves defaults of the form $A : B \,/\, C$ for finite subsets A, B, and C of $\mathcal{L}$, which is intended to represent that one should deduce C if one knows A, given that B is consistent with what is known.

Flooded river example

We illustrate the techniques using an example, which is based on a real application studied by Damien Raclot and Christian Puech [23,24] (see also other work, from the REVIGIS project, on this topic: [37,15,17,2]). An area of land surrounding a flooded river is analysed using aerial photographs and other sources of information, such as elevation models. It is divided up into n parcels of land, or *compartments*, which are small enough so that it can be assumed that the water level is constant within a compartment. Each of these compartments is either partially or completely flooded. Let x_i be the water level (in decimetres above a fixed base level) of compartment i.

The purpose of the analysis is to deduce information about the levels x_i for various compartments i. Expert analysis of the aerial photographs, in conjunction with the other sources of information, generates constraints of the following forms: *upper bounds* of the form $x_i \leq s$, and *lower bounds* of the form: $x_i \geq r$ (where r and s are given numbers) and simple linear constraints of the form $x_j \geq x_i$, which we call a *flow*, since it corresponds to a flow of water from compartment j down to compartment i, which is observable in the photograph. For example, a lower bound can arise from knowledge of the elevation of a flooded compartment, and an upper bound through an observation that a vine is partially submerged. Both types of information (bounds and flows) are uncertain, but the flows are considered as less uncertain than the bounds.

Ignoring the uncertainty, this is a special kind of Simple Temporal Problem (Dechter et al, 1991) [7] (though the variables are spatial rather than temporal, and the variables are state variables as opposed to decision variables); a simple linear time algorithm can be used involving both upstream and downstream propagations (Raclot and Puech, 2003; Wilson, 2002) [24,33], to test consistency and generate inferred bounds on the variables. However, this is not so useful on its own since the input information in the application may well be inconsistent, because e.g., of mis-estimation of elevations.

2.2 A General Class of Monotonic Logics

The initial motivation of this work was to extend various uncertainty formalisms to linear constraints. However, our approaches in Sections 3, 4 and 5 apply to much more general logics, which we refer to here as “monotonic model-theoretic logics”. These include many classical logics, and also logics which allow disjunctions, negations and conjunctions of linear constraints, as well as non-linear constraints. In particular, allowing disjunctions of linear constraints (Koubarakis, 2001) [20] greatly increases the expressive power, for example, allowing disjunctive temporal problems to be represented (Dechter et al.) [7], or allowing non-convex spatial polygons.

Formally, we define a monotonic model-theoretic logic to be a triple $\langle \mathcal{L}, \mathcal{M}, \models \rangle$, where $\mathcal{L}$ and $\mathcal{M}$ are sets and $\models \subseteq \mathcal{M} \times \mathcal{L}$ is a relation between them. $\mathcal{L}$ is called the *language* and $\mathcal{M}$ is called the set of *models*. Relation $\models$ is used to build a semantic entailment relation (which is also called $\models$) between subsets of the language. We use similar definitions as those in Section 2.1 for this more general situation. For model $x \in \mathcal{M}$ and $a \in \mathcal{L}$ we say that x *satisfies* a if $x \models a$ (i.e., if $(x, a) \in \models$). For $A \subseteq \mathcal{L}$ we say that x *satisfies* A, written $x \models A$, if x satisfies every element of A. For subsets A and B of the language $\mathcal{L}$ we say that $A \models B$ if x satisfies (every element of) B for any model x which satisfies A. If $b \in \mathcal{L}$ we sometimes write $A \models b$ to mean $A \models \{b\}$. For $A \subseteq \mathcal{L}$ we write $[A]$ for the set of models that satisfy A, i.e., $\{x \in \mathcal{M} : x \models A\}$. Hence $A \models B$ if and only if $[A] \subseteq [B]$. We say that A is *consistent* if it has a model, i.e., if $[A]$ is non-empty; otherwise A is *inconsistent*.

The relation $\models$ between subsets of the language is reflexive, monotonic and transitive. Specifically, if $A' \subseteq A \subseteq \mathcal{L}$ then $A \models A'$. If $A' \models B$ and $A' \subseteq A$ then $A \models B$. And, if $A \models B$ and $B \models C$ then $A \models C$.

We will sometimes also assume that $\mathcal{L}$ contains an inconsistent element $\bot$, so that $[\bot] = \emptyset$. (This is not a restrictive assumption, since if $\mathcal{L}$ does not contain such an element, we can add one to $\mathcal{L}$.) Then, A ($\subseteq \mathcal{L}$) is inconsistent if and only if $A \models \bot$.

3 Extending Possibilistic Logic

In this section it is shown how Possibilistic Logic (Dubois, Lang and Prade, 94; Dubois and Prade, 2004) [10,12] can be extended to deal with linear constraints, and the other more general logics we are considering. In possibility theory (Dubois and Prade, 1988) [11], degrees of certainty—which are called 'necessity'—are assumed to be totally ordered and representable by numbers in $[0, 1]$; a necessity value of 1, for a proposition, means that the proposition is considered completely certain; a value of 0 means no certainty at all. If the necessity of a is greater than the necessity of b, then a is considered to be better supported by our information than b is. Possibilistic logic involves reasoning about the degrees of necessity of different propositions of interest. In (Standard) Possibilistic Logic, the lower bound of the necessity value of each of a set of propositions is given; from these we wish to deduce the implied (lower bounds for) necessity values of further propositions of interest. For more details about Possibilistic Logic, see [10,14,21,12] and other papers referenced in the survey (Dubois and Prade, 2004) [12].

Possibility distributions, measures and necessity measures. Let Ω be a (finite or infinite) set, representing a mutually exclusive and exhaustive set of possibilities. A possibility distribution on Ω is defined to be a function $\pi : \Omega \to [\,0,1\,]$. The associated possibility measure $\mathrm{Poss}_\pi : 2^\Omega \to [\,0,1\,]$ is given by $\mathrm{Poss}_\pi(X) = \sup\{\pi(\omega) : \omega \in X\}$. The associated necessity measure $\mathrm{Nec}_\pi : 2^\Omega \to [\,0,1\,]$ is given by $\mathrm{Nec}_\pi(X) = 1 - \mathrm{Poss}_\pi(\Omega - X)$. This is intended to represent degrees of support for subsets of Ω. Note that we are considering unnormalised possibility distributions, possibility measures and necessity measures, i.e., we are *not* assuming that $\sup_{\omega\in\Omega} \pi(\omega) = 1$, or $\mathrm{Poss}(\Omega) = 1$ or that $\mathrm{Nec}(\emptyset) = 0$.

Possibility measures and necessity measures on logical language $\mathcal{L}$. We consider any monotonic model-theoretic logic $\langle \mathcal{L}, \mathcal{M}, \models \rangle$, as defined in Section 2.2. A possibility distribution π on $\mathcal{M}$ induces a possibility measure and a necessity measure on $2^{\mathcal{M}}$, which induce values of possibility and necessity for $\mathcal{L}$ by the semantics. We define $\mathrm{Nec}_\pi(a) = \mathrm{Nec}_\pi([a])$ and $\mathrm{Poss}_\pi(a) = \mathrm{Poss}_\pi([a])$, for $a \in \mathcal{L}$. (Similarly, we could define $\mathrm{Nec}_\pi(A) = \mathrm{Nec}_\pi([A])$ for subsets A of $\mathcal{L}$.)

We are interested in statements of the form $\mathrm{Nec}(a) \geq \alpha$, which we abbreviate to (a, α), where $a \in \mathcal{L}$ and $\alpha \in [\,0,1\,]$. Such a pair is called a *necessity-valued formula (over $\mathcal{L}$)* (Dubois et al., 1994) [10]. It gives a lower bound on the degree of support for a. We assume a set of such pairs, and we will deduce further pairs from this, implying which elements of the language are best supported. A set $\mathcal{A}$ of necessity-valued formulae is called a *necessity-valued knowledge base (over $\mathcal{L}$)*. $\mathcal{A}$ can be thought of as an imprecise specification of a necessity measure, and therefore constrains the associated (unknown) possibility distribution $\pi : \mathcal{M} \to [\,0,1\,]$. Possibility distribution π is said to satisfy a necessity-valued formula (a, α) if and only if its associated necessity measure Nec_π satisfies $\mathrm{Nec}_\pi(a) \geq \alpha$. We write in this case that $\pi \models (a, \alpha)$. As in the usual possibilistic logic, we have a simpler characterisation of this condition, which is easily proved.

Lemma 1 *Possibility distribution π satisfies necessity-valued formula (a, α) if and only if $\pi(x) \leq 1 - \alpha$ for all $x \in \mathcal{M}$ such that $x \not\models a$.*

Proof $\pi \models (a, \alpha)$ if and only if $\mathrm{Nec}_\pi(a) \geq \alpha$ which is if and only if $\mathrm{Poss}_\pi(\mathcal{M} - [a]) \leq 1 - \alpha$ which is if and only if $\sup\{\pi(x) : x \in \mathcal{M} - [a]\} \leq 1 - \alpha$, i.e., $\sup\{\pi(x) : x \not\models a\} \leq 1 - \alpha$. This holds if and only if $\pi(x) \leq 1 - \alpha$ for all x such that $x \not\models a$. ■

We say that π satisfies necessity-valued knowledge base $\mathcal{A}$ if and only if π satisfies each of the necessity-valued formulae in $\mathcal{A}$. The entailment relation for this possibilistic logic is defined as follows: we say that $\mathcal{A}$ entails pair (b, β), written $\mathcal{A} \models (b, \beta)$, if and only if $\pi \models (b, \beta)$ for all π such that $\pi \models \mathcal{A}$. In other words, $\mathcal{A} \models (b, \beta)$ if and only if the necessity constraints corresponding to $\mathcal{A}$ imply that $\text{Nec}(b) \geq \beta$.

Let $\mathcal{A}$ be a necessity-valued knowledge base over $\mathcal{L}$, and let $\alpha \in [\,0, 1\,]$. Define the α-cut $\mathcal{A}_\alpha$ to be $\{(a, \gamma) \in \mathcal{A} \,:\, \gamma \geq \alpha\}$, the set of necessity-valued formulae whose necessity is at least α. We also define, $\mathcal{A}^*_\alpha$ to be the classical projection of the α-cut, so that $a \in \mathcal{A}^*_\alpha$ if and only if there exists a pair (a, γ) in $\mathcal{A}$ for some $\gamma \geq \alpha$.

Given a necessity-valued knowledge base $\mathcal{A}$ we would like procedures that enable us to deduce necessity-value formulae which are consequences of $\mathcal{A}$. In addition, given $b \in \mathcal{L}$, we would like to be able to determine the best lower bound on the necessity of b that $\mathcal{A}$ implies; that is, we'd like to compute $Val_{\mathcal{A}}(b)$, which is defined to be $\sup\{\beta \in [\,0, 1\,] \,:\, \mathcal{A} \models (b, \beta)\}$ [10]. $Val_{\mathcal{A}}(b)$ can be considered as the implied necessity of b given $\mathcal{A}$. We have the following key result for this possibilistic logic, which connects entailment in the possibilistic logic with entailment in the underlying (e.g., linear constraints) logic, via the use of classical projections of α-cuts.

Theorem 2 *Let $\mathcal{A}$ be necessity-valued knowledge base over $\mathcal{L}$, and $b \in \mathcal{L}$.*

(i) $\mathcal{A} \models (b, \beta)$ *if and only if for all* $\gamma < \beta$, $\mathcal{A}^*_\gamma \models b$*;*
(ii) *if* $\mathcal{A}$ *is finite then* $\mathcal{A} \models (b, \beta) \iff \mathcal{A}^*_\beta \models b$*;*
(iii) $Val_{\mathcal{A}}(b) = \sup\{\gamma \,:\, \mathcal{A}^*_\gamma \models b\}$*;*
(iv) $\mathcal{A} \models (b, Val_{\mathcal{A}}(b))$*;*
(v) $\mathcal{A} \models (b, \beta)$ *if and only if* $\beta \leq Val_{\mathcal{A}}(b)$*.*

Hence for linear constraints, by finite completeness (Theorem 1), we have that finite $\mathcal{A}$ entails (b, β) if and only if $\mathcal{A}^*_\beta \vdash b$.

Proof (i)(a) First assume $\mathcal{A} \models (b, \beta)$, and, to prove a contradiction, suppose that there exists $\gamma < \beta$ with $\mathcal{A}^*_\gamma \not\models b$. So there exists a model x' with $x' \models \mathcal{A}^*_\gamma$ but $x' \not\models b$. Define possibility distribution π on $\mathcal{M}$ by $\pi(x') = 1 - \gamma$, and for all $x \in \mathcal{M} - \{x'\}$, $\pi(x) = 0$. Now, $\pi(x') \not\leq 1 - \beta$, so, using Lemma 1, $\pi \not\models (b, \beta)$. Consider any $(a, \alpha) \in \mathcal{A}$. To prove that $\pi \models (a, \alpha)$ it is sufficient, by Lemma 1, to show that if $x \in \mathcal{M}$ is such that $\pi(x) > 1 - \alpha$ then $x \models a$. If $x \in \mathcal{M}$ is such that $\pi(x) > 1 - \alpha$ then $x = x'$, which implies that $1 - \gamma > 1 - \alpha$ and so $\alpha > \gamma$; therefore $a \in \mathcal{A}^*_\gamma$, and so, since $x' \models \mathcal{A}^*_\gamma$, we have $x' \models a$, i.e., $x \models a$.

We've shown that $\pi \models (a, \alpha)$ for all $(a, \alpha) \in \mathcal{A}$ and therefore $\pi \models \mathcal{A}$. This, together with $\pi \not\models (b, \beta)$, implies that $\mathcal{A} \not\models (b, \beta)$, which contradicts our initial assumption. Hence $\mathcal{A} \models (b, \beta)$ implies that for all $\gamma < \beta$, $\mathcal{A}^*_\gamma \models b$.

(i)(b) Now assume that for all $\gamma < \beta$, $\mathcal{A}^*_\gamma \models b$. We need to show that $\mathcal{A} \models (b, \beta)$. Consider any possibility distribution π and $x \in \mathcal{M}$ such that $\pi \models \mathcal{A}$

and any $x \not\models b$. We will show that $\pi(x) \leq 1-\beta$ proving, by Lemma 1, that $\pi \models (b, \beta)$ and hence that $\mathcal{A} \models (b, \beta)$, proving the result.

Consider any γ with $\gamma < \beta$. Since $x \not\models b$ and $\mathcal{A}^*_\gamma \models b$, we have $x \not\models \mathcal{A}^*_\gamma$, so there exists $a_\gamma \in \mathcal{A}^*_\gamma$ with $x \not\models a_\gamma$, and so there exists a pair (a_γ, α) in $\mathcal{A}$ for some $\alpha \geq \gamma$. Using the fact that $\pi \models \mathcal{A}$, and hence $\pi \models (a_\gamma, \alpha)$, it follows using Lemma 1 that $\pi(x) \leq 1-\alpha$, and so $\pi(x) \leq 1-\gamma$.

Hence $\pi(x) \leq 1-\gamma$ for all $\gamma < \beta$, and so $\pi(x) \leq s$ for all $s > 1-\beta$, which implies that $\pi(x) \leq 1-\beta$, as required.

(ii) If $\mathcal{A}^*_\beta \models b$ then for any $\gamma < \beta$, $\mathcal{A}^*_\gamma \models b$ (since $\mathcal{A}^*_\gamma \supseteq \mathcal{A}^*_\beta$) so, by part (i), $\mathcal{A} \models (b, \beta)$.

Conversely, suppose that $\mathcal{A}$ is finite, and that $\mathcal{A} \models (b, \beta)$. Since $\mathcal{A}$ is finite we can choose γ such that, for any pair (a, α) with $\alpha < \beta$, we have $\alpha < \gamma < \beta$. Then $\mathcal{A}^*_\gamma = \mathcal{A}^*_\beta$ since, for $(a, \alpha) \in \mathcal{A}$, $[\alpha \geq \beta \iff \alpha \geq \gamma]$. Part (i) implies that $\mathcal{A}^*_\gamma \models b$, i.e., $\mathcal{A}^*_\beta \models b$.

(iii)(a) Let $\hat{b} = \sup\{\gamma : \mathcal{A}^*_\gamma \models b\}$. Consider any $\gamma < \hat{b}$. Then, by definition of $\hat{b}$, there exists γ' such that $\gamma < \gamma' \leq \hat{b}$ and $\mathcal{A}^*_{\gamma'} \models b$. Since $\mathcal{A}^*_\gamma \supseteq \mathcal{A}^*_{\gamma'}$ this implies that $\mathcal{A}^*_\gamma \models b$. So we have for all $\gamma < \hat{b}$, $\mathcal{A}^*_\gamma \models b$, and part (i) implies that $\mathcal{A} \models (b, \hat{b})$. This proves that $\hat{b} \leq \mathit{Val}_{\mathcal{A}}(b)$.

(iii)(b) Conversely, consider any β with $\beta < \mathit{Val}_{\mathcal{A}}(b)$. Choose any β' with $\beta < \beta' < \mathit{Val}_{\mathcal{A}}(b)$. By definition of $\mathit{Val}_{\mathcal{A}}(b)$, we have $\mathcal{A} \models (b, \beta')$ and so, by part (i), $\mathcal{A}^*_\beta \models b$. Hence $\beta \leq \sup\{\gamma : \mathcal{A}^*_\gamma \models b\} = \hat{b}$. Since this holds for any $\beta < \mathit{Val}_{\mathcal{A}}(b)$, we have $\mathit{Val}_{\mathcal{A}}(b) \leq \hat{b}$.

(iv) In proving (iii) we showed that $\mathcal{A} \models (b, \hat{b})$. Hence, by (iii), $\mathcal{A} \models (b, \mathit{Val}_{\mathcal{A}}(b))$.

(v) By definition of $\mathit{Val}_{\mathcal{A}}(b)$, we have $\mathcal{A} \models (b, \beta)$ implies $\beta \leq \mathit{Val}_{\mathcal{A}}(b)$. Conversely, by (iv), $\mathcal{A} \models (b, \mathit{Val}_{\mathcal{A}}(b))$, so if $\beta \leq \mathit{Val}_{\mathcal{A}}(b)$ then $\mathcal{A} \models (b, \beta)$.

■

Consider a given necessity-valued knowledge base $\mathcal{A}$. Suppose that we are interested in finding information about the necessity degree of an element b of the language $\mathcal{L}$. Then, as in the proof of Proposition 13 of (Lang, 2000) [21], we can use a binary search over values of necessity to find increasingly large values $\beta \in [0, 1]$ with $\mathcal{A}^*_\beta \models b$ and hence $\mathcal{A} \models (b, \beta)$. Each value of β involves the checking of an inference in the language; computational efficiency of the procedure is therefore closely tied to the efficiency of deduction in the underlying logic (e.g., in the linear constraints case, this depends on the class of linear constraints used). For the finite case, the procedure will terminate with the maximal value $Val_{\mathcal{A}}(b)$ of β; for the case of infinite $\mathcal{A}$, the procedure may have to terminate before it has found the maximal value. (However, in certain cases with infinite $\mathcal{A}$, analytic optimisation techniques can be used to compute $Val_{\mathcal{A}}(b)$ exactly.) Of particular interest, for given $b \in \mathcal{L}$, is if we can find β with $\mathcal{A} \models (b, \beta)$ but $\mathcal{A} \not\models (\perp, \beta)$, as this indicates positive support for b. Focusing on the set of b with positive support leads to a possibilistic approach to belief revision for such a monotonic model-theoretic underlying logical language, see e.g., (Dubois and Prade, 2004) [12].

Note that, in contrast with (Dubois, Lang and Prade, 94) [10], we allow a necessity-valued knowledge base to be infinite. This is natural in certain situations for representing continuously graded knowledge bases. Not all results for the finite case hold for the infinite case. For example, we have the following result, which is Proposition 11 of [10] in our more general context:

For finite $\mathcal{A}$ we have $\mathcal{A} \models (b, \beta)$ if and only if $\mathcal{A}_\beta \models (b, \beta)$. (This follows from Theorem 2(ii), since $\mathcal{A}^*_\beta = (\mathcal{A}_\beta)^*_\beta$.)

However, this result does not hold in general for infinite $\mathcal{A}$. For example, let $\mathcal{A} = \{(a, \alpha) : 0 < \alpha < 1\}$, where a is some non-tautologous element of $\mathcal{L}$ (e.g., $x_1 \geq 3$). Then $\mathcal{A} \models (a, 1)$. But the α-cut $\mathcal{A}_1$ is the empty set, so $\mathcal{A}_1 \not\models (a, 1)$.

Possibilistic constraints for the flooded river example

We can assign necessity values to the various bounds and flows in the flooding problem. Our inputs then consist of a set $\mathcal{A}$ of pairs (a, α) where α is a lower bound on the necessity of a, and a is either a flow, a lower bound or an upper bound.

For example, suppose a hut is observed partially submerged in compartment 49, giving us upper bound information regarding x_{49}, the water level in that compartment. Our expert is confident (based on the digital elevation model for the ground level, and on an estimate of the height of the hut) that the elevation of the top of the hut is at most 120 decimeters (above the base level). If "confident" is taken to correspond to a necessity value of 0.6, then this gives rise to a statement $\text{Nec}(x_{49} \leq 120) \geq 0.6$, and hence a necessity-valued formula $((x_{49} \leq 120), 0.6)$. The expert tentatively believes—corresponding to necessity value 0.4—that the top of the hut's elevation is at most 110 decimeters, leading to pair $((x_{49} \leq 110), 0.4)$. A flow is observed from compartment 49 down to compartment 86; and this is considered as very reliable information, corresponding to a necessity value of 0.9. This gives rise to the necessity-valued formula $((x_{86} \leq x_{49}), 0.9)$. We can then deduce upper bound information on the level x_{86} of the water in compartment 86: $\text{Nec}(x_{86} \leq 120) \geq 0.6$ and $\text{Nec}(x_{86} \leq 110) \geq 0.4$.

For any α appearing in some pair in the necessity-valued knowledge base $\mathcal{A}$, we could compute, using the linear propagation algorithm, the bounds on each compartment level x_i implied from $\mathcal{A}^*_\alpha$. All these bounds then have necessity value at least α. Given that $\mathcal{A}$ is finite, applying this approach for each α appearing in $\mathcal{A}$ will then give us the implied necessity value for each inferred bound.

An alternative approach is to adapt the propagation algorithm for the constraints to also propagate the necessities. The propagation of the bounds is based on inferences of the form: *From* lower bound $x_i \geq r$ *and* flow constraint $x_j \geq x_i$ *deduce* lower bound $x_j \geq r$. Similarly, for the possibilistic constraints, we can chain a lower bound pair $(x_i \geq r, \alpha)$ and a flow pair $(x_j \geq x_i, \beta)$ to get a lower bound pair $(x_j \geq r, \min(\alpha, \beta))$. This approach generalises easily to the lattice-valued possibilistic logic and assumption-based reasoning approaches described in the next section. The propagation algorithms can be adapted in this way to compute the implied necessity values of lower and upper bounds. We can also determine $\hat{\perp}$, the deduced necessity of $\perp$.

The output of such an approach would be a set of upper bounds and lower bounds for each compartment variable, where each of these bounds has an associated necessity grade, and stronger bounds are associated with smaller necessity values. The strongest bounds with necessity values greater than $\hat{\perp}$ can therefore be considered as constraining the 'best guesses' for the water levels in the compartments. The weaker bounds, with higher necessity values, give us information we can be more confident in.

When dealing with continuous spatial or temporal variables, it can be natural to use a continuous representation of the uncertainty (which could be based on smooth interpolation of discrete information). For example, an expert might indicate, using a graphical user interface, that $\mathrm{Nec}(x_{49} \leq 120 - 50\lambda) \geq 0.6 - \lambda$ for all $\lambda \in [0, 0.6]$ (continuously extending the upper bound information on x_{49} expressed above). This gives rise to an infinite necessity-valued knowledge base: $\{((x_{49} \leq 120 - 50\lambda), 0.6 - \lambda) \,:\, \lambda \in [0, 0.6]\}$.

4 Lattice-Valued Possibilistic Logic and Assumption-based Reasoning

Possibilistic logic can be generalised to a situation where the values of necessity (i.e. the degrees of support) are in a distributive lattice (Dubois, Lang and Prade, 91; 94) [9,10]; see also (Dubois and Prade, 2004) [12]. Degrees of support which are only partially ordered occur naturally in many situations. For example, in the temporal situation described in (Dubois, Lang and Prade,

91) [9], or when there are independent sources of support with incomparable strengths. The grades of support could represent sets of scenarios; in that case, a has generalised necessity degree of at least α if a holds in the set of scenarios represented by α. Closely related partially ordered possibilistic logic systems (based on standard underlying logics) include those described in (Benferhat and Prade, 2005) [3] and in Section 2.3 of (Wilson, 2006) [35].

We first consider extending lattice-valued possibilistic logic to situations where the underlying logic is more general, that is, a monotonic model-theoretic logic $\langle \mathcal{L}, \mathcal{M}, \models \rangle$, as defined in Section 2.2. In Section 4.2 we go on to consider a strongly related system of assumption-based reasoning.

4.1 Lattice-Valued Possibilistic Logic

Let $\mathcal{K} = (K, 0, 1, \bigwedge, \bigvee)$ be a completely distributive lattice (Davey and Priestley, 2002) [5], with greatest lower bound operation $\bigwedge$ and least upper bound operation $\bigvee$ on subsets of K. The associated partial order $\preceq$ on K is given in the usual way: $\alpha \preceq \beta$ if and only if $\alpha \wedge \beta$ (i.e., $\bigwedge \{\alpha, \beta\}$) $= \alpha$.

Define the language $\mathcal{P}$ of this lattice-valued possibilistic logic to consist of all pairs (A, α), where A is a subset of the language $\mathcal{L}$ and $\alpha \in K$ is a lattice element. For example, with the linear constraints language defined in Section 2.1, set A is interpreted as meaning that the true value x of the vector of real valued variables satisfies each constraint in A. The values in the lattice K might be interpreted as truth values (or, alternatively, degrees of preference). The interpretation of (A, α) is then: the truth value of "x satisfies A" is at least α. Extending Standard Possibilistic Logic, we define the semantics is terms of generalised possibility distributions. In the standard case (Section 3), possibility distribution π satisfies pair (A, α) if and only if $1 - \pi(x) \geq \alpha$ for all x such that $x \not\models A$. However, we do not generally have an operation corresponding to $1 - (\cdot)$ within the lattice. To solve this problem we define a complementary scale for the possibility values.

Let $K \mapsto K^*$ be a bijection between K and some set K^*, with α^* being the image of α, and define $(\alpha^*)^* = \alpha$. Generalised possibility distributions π are defined to be functions from $\mathcal{M}$ to K^*. We say π satisfies (A, α) if for all $x \in \mathcal{M}$ such that $x \not\models A$, $(\pi(x))^* \succeq \alpha$. (To recover the usual definitions of possibility distribution etc., we can set $K = [\,0, 1\,]$ with the usual ordering, and set $K^* = [\,0, 1\,]$ and $\alpha^* = 1 - \alpha$; cf. Lemma 1.) For $\Delta \subseteq \mathcal{P}$ and $(B, \beta) \in \mathcal{P}$ this gives the semantic consequence relation:

$\Delta \models (B, \beta)$ if and only if π satisfies (B, β) for all π such that π satisfies (every pair in) Δ.

Theorem 2 cannot be generalised to the lattice case (at least in a simple way) because of the potentially more complex structure of the lattice (in particular it being generally only partially ordered). However, we can still define a very simple sound and complete proof theory.

Proof theory.

From (A, α) *deduce* (B, β) for all (B, β) such that $A \models B$ and $\beta \preceq \alpha$.

From $\{(A, \alpha_i) : i \in I\}$ *deduce* $(A, \bigvee_{i \in I} \alpha_i)$.

From $\{(A_i, \alpha_i) : i \in I\}$ *deduce* $(\bigcup_{i \in I} A_i, \bigwedge_{i \in I} \alpha_i)$.

Let Δ be a subset of $\mathcal{P}$. Define the set of syntactic consequences $\mathcal{C}(\Delta)$ of Δ to be the the intersection of all sets Γ ($\subseteq \mathcal{P}$) (which is the unique smallest set Γ) such that (i) $\Gamma \supseteq \Delta$ and (ii) Γ is closed under the inference rules (i.e., if Γ contains an instance of the left hand side of an inference rule then it contains the corresponding instance of the right hand side). We then define the syntactic consequence relation $\vdash$ by $\Delta \vdash (B, \beta)$ if and only if $(B, \beta) \in \mathcal{C}(\Delta)$. This leads to the following completeness result, which is proved in Section 4.3.

Theorem 3 *[Soundness and Completeness of Paired System] Let* $(B, \beta) \in \mathcal{P}$ *be a pair and let* $\Delta \subseteq \mathcal{P}$ *be a set of pairs. Then* $\Delta \models (B, \beta)$ *if and only if* $\Delta \vdash (B, \beta)$.

When Δ is finite the proof theory can be written in a simpler way, with the second and third inference rules being replaced by:

From (A, α) *and* (A, β) *deduce* $(A, \alpha \vee \beta)$.

From (A, α) *and* (B, β) *deduce* $(A \cup B, \alpha \wedge \beta)$.

Also the definition of syntactic consequence simplifies to the usual kind of definition: $\Delta \vdash (B, \beta)$ if and only if (B, β) can be proved (in a finite number of steps) from Δ using the inference rules.

Even if Δ is infinite, if distributive lattice $\mathcal{K}$ is finite then we can rewrite Δ as the equivalent, but finite, set of pairs $\Delta' = \{(A^\alpha, \alpha) : \alpha \in K\}$, where, for given $\alpha \in K$, set A^α is the union of A over all $(A, \alpha) \in \Delta$, so we again could use the latter more usual (finitary) kind of proof theory.

Even if Δ is infinite, if distributive lattice $\mathcal{K}$ is finite then we can rewrite Δ as the equivalent, but finite, set of pairs $\Delta' = \{(A^\alpha, \alpha) : \alpha \in K\}$, where, for given $\alpha \in K$, set A^α is the union of A over all $(A, \alpha) \in \Delta$, so we again could use the latter more usual (finitary) kind of proof theory.

We can produce a related framework and achieve similar results for systems of pairs which may be viewed as generalised versions of Assumption-Based Truth Maintenance Systems (de Kleer, 1986) [6]. Consider a finite system of pairs of the form (A, ϕ) where A is a subset of the language, and ϕ is a formula in some propositional language $\mathcal{R}$; formula ϕ is intended to represent conditions under which constraints (or formulae) A are known to hold. For example, if an expert tells us that $x_1 \geq 120$ then we can construct a pair $(\{x_1 \geq 120\}, p_1)$, where propositional symbol p_1 represents that the expert is being reliable here. To express logical relationships between these conditions, it can be useful also to allow an additional set of formulae $T \subseteq \mathcal{R}$, which are assumed to be true.

We define the semantics for this assumption-based reasoning systems as follows. Let Ω be the set of $\mathcal{R}$-valuations satisfying T. Models are defined to be pairs (x, ω) for $x \in \mathcal{M}$ and $\omega \in \Omega$. Pair (x, ω) represents a possible assignment to the propositional variables and also a model of the language $\mathcal{L}$ (which is an assignment to all the real-valued variables in the linear constraints case). Pair (A, ϕ) restricts possible models (x, ω), and is intended to represent that, if condition ϕ holds, then all of A hold; we therefore say that (x, ω) satisfies (A, ϕ) if the following condition holds: [ω satisfies ϕ] implies $x \models A$. Hence (A, ϕ) can be thought of as an implication: *if ϕ holds then A holds.* As usual we extend this to a semantic consequence relation on pairs: $\Delta \models (A, \phi)$ if (A, ϕ) is satisfied by all (x, ω) satisfying every element of Δ.

Define syntactic entailment $\vdash$ using the following proof theory:

From (A, ϕ) deduce (B, ψ) for all (B, ψ) in $\mathcal{P}$ such that $A \models B$ and $T \cup \{\psi\} \models \phi$.

From (A, ϕ) and (A, ψ) deduce $(A, \phi \vee \psi)$.

From (A, ϕ) and (B, ψ) deduce $(A \cup B, \phi \wedge \psi)$.

Again, this simple proof theory is sound and complete, as proved in Section 4.4.

Theorem 4 (Completeness of Assumption-based System.) *With the above proof theory, finite Δ syntactically entails (B, ψ) if and only if Δ semantically entails (B, ψ).*

Given set of pairs Δ, one can associate with a set $B \subseteq \mathcal{L}$ a formula ϕ_B in $\mathcal{R}$ which expresses precisely the conditions under which B can be deduced; that is, $\Delta \models (B, \phi)$ if and only if $T \cup \{\phi\} \models \phi_B$ (so that ϕ_B is a logically weakest formula such that (B, ϕ_B) is deducible from Δ). If one had a probability measure on $\mathcal{R}$, satisfying $\Pr(T) = 1$, then this can be used to generate the probability that B can be deduced, i.e., $\Pr(\phi_B)$, which can be considered as a degree of belief in B. An important special case is where Δ can be written as $\{(A_i, p_i) : i = 1, \ldots, m\}$ where each p_i is a propositional variable, and $T = \emptyset$. This is an assumption-based system. If each p_i is probabilistically independent of the others, and has a chance r_i of holding, this generates a probability measure on $\mathcal{R}$ and hence degrees of belief. This situation corresponds to a special case of the generalised Dempster-Shafer theory described in Section 5, and is strongly related to work on probabilistic argumentation systems (Haenni et al., 2000; Anrig et al., 1997) [13,1].

4.3 Proving Theorem 3

We consider a fixed (possibly infinite) set of pairs $\Delta \subseteq \mathcal{P}$ which we label as $\{(A_i, \alpha_i) : i \in I\}$. Define, for $x \in \mathcal{M}$, $I_x = \{i \in I : x \not\models A_i\}$. Define model π_Δ of the lattice-valued possibilistic logic by, for $x \in \mathcal{M}$, $(\pi_\Delta(x))^* = \bigvee_{i \in I_x} \alpha_i$.

The first three lemmas establish Proposition 1. Lemmas 5 and 6 then establish Proposition 2; the completeness property follows immediately from these two propositions. Soundness follows from Lemma 7.

Lemma 2 *π_Δ satisfies (A, α) if and only if for all $x \not\models A$, $\bigvee_{i \in I_x} \alpha_i \succeq \alpha$.*

Proof π_Δ satisfies (A, α) if and only if for all $x \in \mathcal{M}$ such that $x \not\models A$, $(\pi_\Delta(x))^* \succeq \alpha$. i.e., $\bigvee_{i \in I_x} \alpha_i \succeq \alpha$. ■

Lemma 3 *Model π satisfies Δ if and only if for all $x \in \mathcal{M}$, $(\pi(x))^* \succeq \bigvee_{i \in I_x} \alpha_i$, i.e., $(\pi(x))^* \succeq (\pi_\Delta(x))^*$.*

Proof Suppose first that π satisfies Δ. Then π satisfies (A_i, α_i) for any $i \in I$. So, for all $x \in \mathcal{M}$ we have $(\pi(x))^* \succeq \alpha_i$ if $x \not\models A_i$, i.e., if $i \in I_x$. Hence $(\pi(x))^* \succeq \bigvee_{i \in I_x} \alpha_i$.

Conversely, suppose that for all $x \in \mathcal{M}$, $(\pi(x))^* \succeq \bigvee_{i \in I_x} \alpha_i$. If $x \not\models A_i$ then $i \in I_x$, so $(\pi(x))^* \succeq \alpha_i$, which shows that π satisfies every element (A_i, α_i) of Δ, as required. ■

Lemma 4 *π_Δ satisfies Δ.*

Proof This follows immediately from Lemma 3. ■

Proposition 1 *If $\Delta \models (B, \beta)$ then $(\bigwedge_{x \not\models B} \bigvee_{i \in I_x} \alpha_i) \succeq \beta$.*

Proof Suppose $\Delta \models (B, \beta)$. Lemma 4 implies that π_Δ satisfies (B, β), so for all $x \not\models B$, $\bigvee_{i \in I_x} \alpha_i \succeq \beta$, by Lemma 2. Hence β is a lower bound for the infimum of these, i.e., $\bigwedge_{x \not\models B} \bigvee_{i \in I_x} \alpha_i \succeq \beta$. ■

For an arbitrary subset B of $\mathcal{L}$, let $S_B = \{\sigma \subseteq I \,:\, \bigcup_{i \in \sigma} A_i \models B\}$.

Lemma 5 *For any $B \subseteq \mathcal{L}$, $\mathcal{C}(\Delta)$ contains the pair $(B, \bigvee_{\sigma \in S_B} \bigwedge_{i \in \sigma} \alpha_i)$.*

Proof For any $\sigma \subseteq I$, we have that $\mathcal{C}(\Delta)$ contains $\{(A_i, \alpha_i) \,:\, i \in \sigma\}$. So by the third inference rule, $\mathcal{C}(\Delta)$ contains $(\bigcup_{i \in \sigma} A_i, \bigwedge_{i \in \sigma} \alpha_i)$. Therefore, if $\sigma \in S_B$ then $\mathcal{C}(\Delta)$ contains $(B, \bigwedge_{i \in \sigma} \alpha_i)$, using the first inference rule. Hence, by the second inference rule, $\mathcal{C}(\Delta)$ contains $(B, \bigvee_{\sigma \in S_B} \bigwedge_{i \in \sigma} \alpha_i)$, as required. ■

Lemma 6 *For any $B \subseteq \mathcal{L}$, $\bigvee_{\sigma \in S_B} \bigwedge_{i \in \sigma} \alpha_i$ is equal to $\bigwedge_{x \not\models B} \bigvee_{i \in I_x} \alpha_i$.*

Proof A lattice is said to be completely distributive if for any doubly indexed set $\{\alpha_{r,s} \,:\, r \in R, s \in S\}$,

$$\bigwedge_{r \in R} \bigvee_{s \in S} \alpha_{r,s} = \bigvee_{e:R \to S} \bigwedge_{r \in R} \alpha_{r,e(r)},$$

where the $\bigvee$ on the right-hand-side is over all functions e from R to S.

Let $R = \mathcal{M} - [B] = \{x \in \mathcal{M} \,:\, x \not\models B\}$, let $S = I$, and define $\alpha_{x,i} = \alpha_i$ if $i \in I_x$ (i.e., if $x \not\models A_i$), and, otherwise, let $\alpha_{x,i} = 0$, the minimum element of the lattice. Applying the completely distributive property gives

$$\bigwedge_{x \in R} \bigvee_{i \in I} \alpha_{x,i} = \bigvee_{e:R \to I} \bigwedge_{x \in R} \alpha_{x,e(x)}.$$

Now,

$$\bigwedge_{x \in R} \bigvee_{i \in I} \alpha_{x,i} = \bigwedge_{x \not\models B} \bigvee_{i \in I_x} \alpha_i,$$

since elements 0 can be omitted from an application of $\bigvee$. Also,

$$\bigvee_{e:R \to I} \bigwedge_{x \in R} \alpha_{x,e(x)} = \bigvee_{\substack{e:R \to I \\ \forall x, e(x) \in I_x}} \bigwedge_{x \in R} \alpha_{e(x)},$$

because we need only consider functions e such that for all $x \in R$, $e(x) \in I_x$, since other functions e just give rise to an element 0, which can be omitted from the application of $\bigvee$. Using idempotence of $\bigwedge$ and $\bigvee$, this last term can be written as $\bigvee_{\sigma \in Q} \alpha_\sigma$, where Q is a particular set of subsets of I, and for $\sigma \subseteq I$, we define α_σ to be $\bigwedge_{i \in \sigma} \alpha_i$. A set σ is in Q if and only if there exists some function $e : R \to I$ such that $e(x) \in I_x$ for all $x \in R$, and $\sigma = \{e(x) \,:\, x \in R\}$. Putting these parts together we have that $\bigwedge_{x \not\models B} \bigvee_{i \in I_x} \alpha_i$ is equal to $\bigvee_{\sigma \in Q} \alpha_\sigma$. Hence, to complete the proof, we just have to show that $\bigvee_{\sigma \in Q} \alpha_\sigma = \bigvee_{\sigma \in S_B} \alpha_\sigma$.

We will first show that $Q \subseteq S_B$ which implies that $\bigvee_{\sigma \in Q} \alpha_\sigma \preceq \bigvee_{\sigma \in S_B} \alpha_\sigma$. Consider any $\sigma \in Q$. Then for any $x \not\models B$, there exists an element $j = e(x) \in \sigma$

with $j \in I_x$, i.e., $x \not\models A_j$ and hence $x \not\models \bigcup_{i\in\sigma} A_i$; this proves that $\bigcup_{i\in\sigma} A_i \models B$ and hence $\sigma \in S_B$, as required.

We will now show that for any $\sigma \in S_B$ there exists an element $\sigma' \in Q$ with $\sigma' \subseteq \sigma$, and hence $\alpha_{\sigma'} \succeq \alpha_\sigma$, this proves that $\bigvee_{\sigma\in S_B} \alpha_\sigma \preceq \bigvee_{\sigma\in Q} \alpha_\sigma$, and hence $\bigvee_{\sigma\in Q} \alpha_\sigma = \bigvee_{\sigma\in S_B} \alpha_\sigma$.

Consider any $\sigma \in S_B$. By definition of S_B, for any $x \not\models B$, we have $x \not\models \bigcup_{i\in\sigma} A_i$, so there exists $i_x \in \sigma$ such that $x \not\models A_{i_x}$. Define function $e : R \to I$ by for all $x \in R$, $e(x) = i_x$, which is in I_x, and let $\sigma' = \{e(x) : x \in R\}$. By construction, $\sigma' \in Q$ and $\sigma' \subseteq \sigma$. ■

Proposition 2 *If* $(\bigwedge_{x\not\models B} \bigvee_{i\in I_x} \alpha_i) \succeq \beta$ *then* $\mathcal{C}(\Delta)$ *contains the pair* (B, β)*.*

Proof Lemmas 5 and 6 immediately imply that $\mathcal{C}(\Delta)$ contains the pair $(B, \bigwedge_{x\not\models B} \bigvee_{i\in I_x} \alpha_i)$. If $(\bigwedge_{x\not\models B} \bigvee_{i\in I_x} \alpha_i) \succeq \beta$ then, by the first inference rule, $\mathcal{C}(\Delta)$ contains the pair (B, β). ■

Lemma 7 *Let* π *be any function from* $\mathcal{M}$ *to* K^**. Then*

(i) if π *satisfies* (A, α)*, and* (B, β) *is such that* $A \models B$ *and* $\beta \preceq \alpha$ *then* π *satisfies* (B, β)*;*
(ii) if π *satisfies each of* $\{(A, \alpha_i) : i \in I\}$ *then* π *satisfies* $(A, \bigvee_{i\in I} \alpha_i)$
(iii) if π *satisfies each of* $\{(A_i, \alpha_i) : i \in I\}$ *then* π *satisfies* $(\bigcup_{i\in I} A_i, \bigwedge_{i\in I} \alpha_i)$*.*

Proof (i) Suppose that π satisfies (A, α), and (B, β) is such that $A \models B$ and $\beta \preceq \alpha$. Consider any $x \in \mathcal{M}$ such that $x \not\models B$. Therefore $x \not\models A$, and so $(\pi(x))^* \succeq \alpha \succeq \beta$ and hence $(\pi(x))^* \succeq \beta$, proving that π satisfies (B, β).

(ii) Suppose that π satisfies each of $\{(A, \alpha_i) : i \in I\}$. Consider any $x \in \mathcal{M}$ such that $x \not\models A$. Then for all $i \in I$, $(\pi(x))^* \succeq \alpha_i$, so $(\pi(x))^* \succeq \bigvee_{i\in I} \alpha_i$, showing that π satisfies $(A, \bigvee_{i\in I} \alpha_i)$.

(iii) Suppose that π satisfies each of $\{(A_i, \alpha_i) : i \in I\}$. Consider any $x \in \mathcal{M}$ such that $x \not\models \bigcup_{i\in I} A_i$. So there exists some $i \in I$ such that $x \not\models A_i$, which implies that $(\pi(x))^* \succeq \alpha_i$ and so $(\pi(x))^* \succeq \bigwedge_{i\in I} \alpha_i$, proving that π satisfies $(\bigcup_{i\in I} A_i, \bigwedge_{i\in I} \alpha_i)$. ■

Proof of Theorem 3. Soundness: Let $\mathcal{C}'(\Delta)$ be the semantic consequences of Δ, i.e., the set of all pairs (B, β) such that $\Delta \models (B, \beta)$. To prove soundness, it is sufficient to show that $\mathcal{C}'(\Delta)$ contains Δ and is closed under the inference rules, as this implies $\mathcal{C}'(\Delta) \supseteq \mathcal{C}(\Delta)$, and hence $\Delta \vdash (B, \beta)$ implies $\Delta \models (B, \beta)$.

(1) If Δ contains a pair (B, β) then any π which satisfies Δ satisfies (B, β), so $\Delta \models (B, \beta)$. This implies that $\mathcal{C}'(\Delta)$ contains (B, β) and hence contains Δ.

(2) Suppose $\mathcal{C}'(\Delta)$ contains (A, α), and that B and β are such that $A \models B$ and

$\beta \preceq \alpha$. Consider any π such that π satisfies Δ, so π satisfies (A, α). Lemma 7 then implies that π satisfies (B, β), and so $\Delta \models (B, \beta)$ proving that $\mathcal{C}'(\Delta)$ is closed under the first inference rule. Similarly, Lemma 7 implies that $\mathcal{C}'(\Delta)$ is closed under the second two inference rules.

Completeness: Suppose $\Delta \models (B, \beta)$. By Proposition 1, $(\bigwedge_{x \not\models B} \bigvee_{i \in I_x} \alpha_i) \succeq \beta$. Hence by Proposition 2, $\mathcal{C}(\Delta)$ contains the pair (B, β), as required. ■

4.4 Proving Theorem 4

One might prove Theorem 4 using the results of the last section. Here, however, we give a direct proof. We consider again a fixed set of pairs Δ, which we write as $\{(A_i, \phi_i) : i \in I\}$, where I is finite, since we are assuming that Δ is finite. For an arbitrary subset B of $\mathcal{L}$, we let $S_B = \{\sigma \subseteq I : \bigcup_{i \in \sigma} A_i \models B\}$.

The completeness part of the proof follows using the first two lemmas; soundness follows from Lemma 10.

Lemma 8 *For any $B \subseteq \mathcal{L}$, Δ syntactically entails $(B, \bigvee_{\sigma \in S_B} \bigwedge_{i \in \sigma} \phi_i)$.*

Proof The proof of this is very similar to that for Lemma 5. Consider any $\sigma \in S_B$. Since σ is finite, we can repeatedly apply the third inference rule, to show that Δ syntactically entails $(\bigcup_{i \in \sigma} A_i, \bigwedge_{i \in \sigma} \phi_i)$. Since $\sigma \in S_B$, we have $\bigcup_{i \in \sigma} A_i \models B$, so we can use the first inference rule to show that Δ syntactically entails $(B, \bigwedge_{i \in \sigma} \phi_i)$. Hence, by repeated application of the second inference rule, Δ syntactically entails $(B, \bigvee_{\sigma \in S_B} \bigwedge_{i \in \sigma} \alpha_i)$, as required. ■

Lemma 9 *If Δ semantically entails (B, ψ) then $T \cup \{\psi\} \models \bigvee_{\sigma \in S_B} \bigwedge_{i \in \sigma} \phi_i$.*

Proof Suppose that Δ semantically entails (B, ψ), and consider any $\omega \in \Omega$ such that ω satisfies ψ. It is sufficient to show that ω satisfies $\bigvee_{\sigma \in S_B} \bigwedge_{i \in \sigma} \phi_i$.

Consider any $x \in \mathcal{M}$ such that $x \not\models B$. Then (x, ω) does not satisfy (B, ψ), so (x, ω) does not satisfy Δ, by the definition of semantic consequence in the paired system. Hence there exists some $i_x \in I$ such that (x, ω) does not satisfy (A_{i_x}, ϕ_{i_x}), i.e., ω satisfies ϕ_{i_x} but $x \not\models A_{i_x}$.

Define $\sigma_\omega = \{i_x : x \in \mathcal{M}, x \not\models B\}$. If $x \in \mathcal{M}$ is such that $x \not\models B$ then $x \not\models A_{i_x}$ (as shown above) and so $x \not\models \bigcup_{i \in \sigma_\omega} A_i$, which proves that $\bigcup_{i \in \sigma_\omega} A_i \models B$, and hence that $\sigma_\omega \in S_B$. For all $i \in \sigma_\omega$, we have that ω satisfies ϕ_i, so ω satisfies $\bigwedge_{i \in \sigma_\omega} \phi_i$ and hence ω satisfies $\bigvee_{\sigma \in S_B} \bigwedge_{i \in \sigma} \phi_i$, as required. ■

Lemma 10 *Consider an arbitrary pair (x, ω) with $x \in \mathcal{M}$ and $\omega \in \Omega$.*

(i) If (x, ω) satisfies (A, ϕ) then (x, ω) satisfies (B, ψ) if $A \models B$ and $T \cup \{\psi\} \models \phi$.
(ii) If (x, ω) satisfies (A, ϕ) and (A, ψ) then (x, ω) satisfies $(A, \phi \vee \psi)$.
(iii) If (x, ω) satisfies (A, ϕ) and (B, ψ) then (x, ω) satisfies $(A \cup B, \phi \wedge \psi)$.

Proof (i) Suppose (x, ω) satisfies (A, ϕ) and $A \models B$ and $T \cup \{\psi\} \models \phi$. If ω satisfies ψ then ω satisfies ϕ, which implies that $x \models A$ (since (x, ω) satisfies (A, ϕ)). Hence $x \models B$, proving that (x, ω) satisfies (B, ψ).

(ii) Suppose that (x, ω) satisfies (A, ϕ) and (A, ψ). If ω satisfies $\phi \vee \psi$ then either ω satisfies ϕ and so $x \models A$, or ω satisfies ψ and so also then $x \models A$. This proves that (x, ω) satisfies $(A, \phi \vee \psi)$.

(iii) Suppose that (x, ω) satisfies (A, ϕ) and (B, ψ). If ω satisfies $\phi \wedge \psi$ then ω satisfies ϕ and hence $x \models A$, and also, ω satisfies ψ and so $x \models B$. Therefore $x \models A \cup B$ proving that (x, ω) satisfies $(A \cup B, \phi \wedge \psi)$.

■

Proof of Theorem 4. Suppose first that Δ syntactically entails (B, ψ). Consider any pair (x, ω) satisfying Δ. Using Lemma 10, and by induction on the length of the proof, it follows that (x, ω) satisfies (B, ψ). This proves that Δ semantically entails (B, ψ).

Now, conversely, suppose that Δ semantically entails (B, ψ), and so by Lemma 9, $T \cup \{\psi\} \models \bigvee_{\sigma \in S_B} \bigwedge_{i \in \sigma} \phi_i$. Lemma 8 tells us that Δ syntactically entails $(B, \bigvee_{\sigma \in S_B} \bigwedge_{i \in \sigma} \phi_i)$, so we can apply the first inference rule, showing that Δ syntactically entails (B, ψ). ■

5 Extending Dempster-Shafer Theory

This section shows how Dempster-Shafer theory can be extended to reason with linear constraints, and to the other monotonic logic formalisms we consider. See (Kohlas and Monney, 91) [19] for a somewhat related approach to Dempster-Shafer theory for spatial and temporal reasoning, and (Besnard and Kohlas, 95; Kohlas, 2003) [4,18] for other work on generalising Dempster-Shafer theory to logics.

Dempster-Shafer theory, with the view taken here, can be considered as representing situations where we have a probability distribution over a set, and a logical relation between the set and the propositions of interest. For example, in the flooded river problem, one such situation is if we have a probability distribution over the height of a vine, and the vine is only partially covered by the water, implying that the water level there is below the top of the vine. The probabilistic and logical information together can be used to generate degrees of belief in propositions of interest.

5.1 Basic Definitions of Dempster-Shafer Theory

The formalism of (Shafer, 76) [27] was derived from that of Arthur Dempster [8]; Dempster's framework is more convenient for our purposes, and we describe the mathematical basics of a slight variant of it (see (Wilson, 2000) [32]).

Let Θ be a set, which is interpreted as a set of mutually exclusive and exhaustive propositions, or as a set of 'possible worlds'. Θ is known as a *frame [of discernment]*. The propositions of interest are all assumed to be expressed as subsets of the frame. An uncertain piece of information regarding Θ is represented as a *source triple over* Θ, which is defined to be a triple $(\Omega, \mathrm{P}, \Gamma)$ where Ω is a set, P is a strictly positive probability distribution over Ω (so that for all $\omega \in \Omega$, $\mathrm{P}(\omega) \neq 0$) and Γ is a function from Ω to $2^\Theta - \{\emptyset\}$. So we have probabilistic information P about another set Ω, which is related to Θ by Γ. Mapping Γ expresses a logical connection between Ω and Θ: for $\omega \in \Omega$, $\Gamma(\omega)$ is the set of elements of Θ which are compatible with ω. Associated with a source triple S is a *belief function* $\mathrm{Bel}_S : 2^\Theta \to [\,0, 1\,]$ giving "degrees of belief" in subsets of the frame. We define $\mathrm{Bel}_S(X) = \mathrm{P}\{\omega : \Gamma(\omega) \subseteq X\}$, which is the probability that random set $\Gamma(\omega)$ is a subset of X. $\mathrm{Bel}_S(X)$ can be thought of as the probability that X is implied by the piece of uncertain information represented by the source triple. Belief functions are intended as representations of subjective degrees of belief, as described in (Shafer 76; 81) [27,28].

Suppose we have a number of source triples over Θ each representing a separate piece of uncertain information. The combined effect of these, given the appropriate independence assumptions, is calculated using Dempster's rule (of combination). The result of applying Dempster's rule to a finite set of source triples $\{(\Omega_i, \mathrm{P}_i, \Gamma_i)$, for $i = 1, \ldots, k\}$, is defined to be the source triple $(\Omega, \mathrm{P}_{\mathrm{DS}}, \Gamma)$ over Θ, which is defined as follows. Let $\Omega^\times = \Omega_1 \times \cdots \times \Omega_k$. For $\omega \in \Omega^\times$, $\omega(i)$ is defined to be its ith component, so that $\omega = (\omega(1), \ldots, \omega(k))$. Define $\Gamma' \colon \Omega^\times \to 2^\Theta$ by $\Gamma'(\omega) = \bigcap_{i=1}^k \Gamma_i(\omega(i))$ and probability distribution P' on $\Omega^\times$ by $\mathrm{P}'(\omega) = \prod_{i=1}^k \mathrm{P}_i(\omega(i))$, for $\omega \in \Omega^\times$. Let Ω be the set $\{\omega \in \Omega^\times : \Gamma'(\omega) \neq \emptyset\}$, let Γ be Γ' restricted to Ω, and let probability function P_{DS} on Ω be P' conditioned on Ω, so that for $\omega \in \Omega$, $\mathrm{P}_{\mathrm{DS}}(\omega) = \mathrm{P}'(\omega)/\mathrm{P}'(\Omega)$.

The combined measure of belief Bel over Θ is thus given, for $X \subseteq \Theta$, by $\mathrm{Bel}(X) = \mathrm{P}_{\mathrm{DS}}(\{\omega \in \Omega : \Gamma(\omega) \subseteq X\})$.

We consider an underlying monotonic model-theoretic logic $\langle \mathcal{L}, \mathcal{M}, \models \rangle$, as defined in Section 2.2. The definitions of the previous section can be extended easily to such logical languages via the semantics. Essentially, the belief of $A \subseteq \mathcal{L}$ is defined to be the belief of $[A]$.

Define a source triple over $\mathcal{L}$ to be a triple $(\Omega, \mathrm{P}, \Gamma)$ where Ω is a set, P is a strictly positive probability function (i.e., probability density function or probability mass function) on Ω and Γ is a function from Ω to $\mathcal{F}$, where $\mathcal{F}$ is the set of finite[2] consistent subsets of $\mathcal{L}$. One interpretation of source triples is that we're interested in $\mathcal{L}$, but we have Bayesian beliefs about Ω, and a logical connection between the two, expressed by Γ. The interpretation of Γ is that if the proposition represented by ω is true, then the proposition represented by $\Gamma(\omega)$ is also true.

We can associate with a source triple $S = (\Omega, \mathrm{P}, \Gamma)$ over $\mathcal{L}$ a generalised belief function $\mathrm{Bel}_S : \mathcal{L} \to [\,0, 1\,]$ giving degrees of belief in elements in the language $\mathcal{L}$. This is given as follows: for $a \in \mathcal{L}$, $\mathrm{Bel}_S(a) = \mathrm{P}(\{\omega \in \Omega : \Gamma(\omega) \models a\})$ (assuming that this set is measurable), which we abbreviate to $\mathrm{P}(\Gamma(\omega) \models a)$; the belief in a is the probability that a is implied. We can also define Bel for finite subsets A of $\mathcal{L}$ in a similar fashion: $\mathrm{Bel}_S(A) = \mathrm{P}(\Gamma(\omega) \models A)$.

Source triple $S = (\Omega, \mathrm{P}, \Gamma)$ over $\mathcal{L}$ corresponds to a source triple $S_0 = (\Omega, \mathrm{P}, \Gamma_0)$ over the set of models $\mathcal{M}$ (as defined in the last section), where $\Gamma_0 : \Omega \to 2^{\mathcal{M}}$ is given by, for $\omega \in \Omega$, $\Gamma_0(\omega) = [\Gamma(\omega)]$. As one would hope, we then have, for $A \subseteq \mathcal{L}$, $\mathrm{Bel}_S(A) = \mathrm{Bel}_{S_0}([A])$.

Dempster's rule of combination. Dempster's rule of combination can also be easily extended. Suppose we have a number of source triples $(\Omega_i, \mathrm{P}_i, \Gamma_i)$, for $i = 1, \ldots, k$, each representing a separate piece of uncertain information. The combination $(\Omega, \mathrm{P}_{\mathrm{DS}}, \Gamma)$ of these source triples over $\mathcal{L}$ is defined as follows.

As in the last section, let $\Omega^{\times} = \Omega_1 \times \cdots \times \Omega_k$. Define $\Gamma' \colon \Omega^{\times} \to \mathcal{F}$ by $\Gamma'(\omega) = \bigcup_{i=1}^{k} \Gamma_i(\omega(i))$ and probability function P' on $\Omega^{\times}$ by $\mathrm{P}'(\omega) = \prod_{i=1}^{k} \mathrm{P}_i(\omega(i))$, for $\omega \in \Omega^{\times}$. Let Ω be the set $\{\omega \in \Omega^{\times} : [\Gamma'(\omega)] \neq \emptyset\}$, let Γ be Γ' restricted to Ω, and let probability function P_{DS} on Ω be P' conditioned by Ω, so that for $\omega \in \Omega$, $\mathrm{P}_{\mathrm{DS}}(\omega) = \mathrm{P}'(\omega)/\mathrm{P}'(\Omega)$ (given that $\mathrm{P}'(\Omega) \neq 0$).

The combined measure of belief Bel is the belief function associated with the combined source triple, and is thus given, for finite $A \subseteq \mathcal{L}$, by $\mathrm{Bel}(A) =$

[2] We restrict to finite subsets so that for the linear constraints case, we can make use of the finite completeness result, Theorem 1. This restriction could be relaxed.

$\mathrm{P_{DS}}(\Gamma(\omega) \models A)$. For the linear constraints case, by Theorem 1, this equals $\mathrm{P_{DS}}(\Gamma(\omega) \vdash A)$ since $\Gamma(\omega)$ is a finite subset of $\mathcal{L}$.

Computing combined belief. It is possible to adapt some of the standard approaches (see e.g., (Wilson, 2000) [32]) for computing combined belief in this more general situation. In particular, various Monte-Carlo algorithms can be adapted to give arbitrarily close approximations of values of belief. We assume that we have some procedure for determining whether $\Gamma(\omega) \models A$ holds or not.

Since, for finite $A \subseteq \mathcal{L}$, $\mathrm{Bel}(A) = \mathrm{P_{DS}}(\Gamma(\omega) \models A)$, to calculate $\mathrm{Bel}(A)$ we can repeat a large number of trials of a Monte-Carlo algorithm where for each trial, we pick ω with chance $\mathrm{P_{DS}}(\omega)$ and say that the trial succeeds if $\Gamma(\omega) \models A$, and fails otherwise. $\mathrm{Bel}(A)$ is then estimated by the proportion of the trials that succeed. The most straight-forward way is to pick ω with chance $\mathrm{P_{DS}}(\omega)$ by repeatedly (if necessary) picking $\omega \in \Omega^{\times}$ with chance $\mathrm{P}'(\omega)$ until we get an ω in Ω. Picking ω with chance $\mathrm{P}'(\omega)$ is easy: for each $i = 1, \ldots, k$, we pick $\omega_i \in \Omega_i$ with chance $\mathrm{P}_i(\omega_i)$ and let $\omega = (\omega_1, \ldots, \omega_k)$. If the conflict probability $1 - \mathrm{P}'(\Omega)$ is bounded away from 1, this algorithm has low complexity, proportional to the complexity of proof in the logic (Wilson, 1991, 2000) [31,32], but with a high constant factor because of needing a large number of trials to achieve a good estimate of values of belief. If the conflict is very high, we would be better off using more complex Monte-Carlo algorithms, such as a Markov Chain Monte Carlo algorithm (Moral and Wilson, 1994; Wilson, 2000) [22,32].

Dempster-Shafer approach for the flooded river example

An uncertain flow constraint $x_j \geq x_i$ with reliability $p \in [\,0, 1\,]$ can be represented as a source triple $(\{\omega, \omega'\}, \mathrm{P}, \Gamma)$ over a linear constraints language $\mathcal{L}$ (see Section 2.1), with $\mathrm{P}(\omega) = p$, $\mathrm{P}(\omega') = 1 - p$ and $\Gamma(\omega) = \{x_j \geq x_i\}$, and $\Gamma(\omega') = \{\top\}$. This corresponds to a *simple support function* (Shafer, 76) [27]. Given just this source triple, we can deduce the constraint $x_j \geq x_i$ with chance p, so that the associated belief function has $\mathrm{Bel}(\{x_j \geq x_i\}) = p$.

Suppose we have uncertain lower bound information on x_2, the water level in compartment 2, based on information from a digital elevation model and an observation that this compartment is flooded, which is represented by a source triple $(\{\omega_1, \omega_2, \omega_3\}, \mathrm{P}_2, \Gamma_2)$ defined as follows: $\mathrm{P}_2(\omega_1) = 0.5$, $\mathrm{P}_2(\omega_2) = 0.4$ and $\mathrm{P}_2(\omega_3) = 0.1$; and $\Gamma_2(\omega_1) = (x_2 \geq 100)$, $\Gamma_2(\omega_2) = (x_2 \geq 90)$, and $\Gamma_2(\omega_3) = \top$. This leads to a belief function with $\mathrm{Bel}(x_2 \geq 100) = 0.5$ (arising from a judgement that there is 50% chance that the elevation of this compartment is at least 100), and $\mathrm{Bel}(x_2 \geq 90) = 0.5 + 0.4 = 0.9$. If we had just this source triple and an uncertain flow constraint $x_8 \geq x_2$ with reliability 0.95, then the combined belief that $x_8 \geq 100$ would be equal to $0.5 \times 0.95 = 0.475$. Note that a continuously graded representation of an uncertain bound can also be very natural, which requires the use of infinite Ω in the source triple; Monte-Carlo simulation can be still used for computation.

Given a large number of uncertain flow constraints and uncertain upper and lower bounds, we can represent each by source triples and combine them using Dempster's rule, given that the appropriate independence assumptions are satisfied. (It is also possible to model dependencies in the information, by constructing a different combined probability function from P_{DS}.) We can then compute the combined beliefs in constraints of interest, or use a Monte-Carlo algorithm to approximate them. For example, if we find that $\mathrm{Bel}(\{x_6 \geq 120,\ x_6 \leq 130\}) = 0.7$ then it means that with chance 0.7 we can deduce that the level of compartment 6 is in the interval $[120, 130]$; the value 0.7 can be viewed as a kind of lower probability for: $x_6 \in [120, 130]$.

6 Summary

This paper shows how a number of important uncertainty formalisms can be extended to deal with uncertain linear constraints, and other logical formalisms. The formalisms we discuss are possibilistic logic, a lattice-valued possibilistic logic, a general form of assumption-based reasoning, and Dempster-Shafer theory. We show how deductions can be made in this possibilistic logic, and how Dempster-Shafer algorithms can be adapted, and construct sound and complete proof theories for the lattice-valued possibilistic logic, and the assumption-based reasoning formalism.

Acknowledgements

This work was supported by the REVIGIS project, IST-1999-14189, and was partly based on works supported by the Science Foundation Ireland under Grant No. 00/PI.1/C075 and Grant No. 05/IN/I886. It benefitted from the referees' comments and many discussions with my colleagues on the REVIGIS project, in particular, Mahat Khelfallah, Eric Wurbel, Belaid Benhamou, Robert Jeansoulin, Odile Papini, and especially Christian Puech and Damien Raclot.

References

[1] B. Anrig, R. Haenni, J. Kohlas, and N. Lehmann. Assumption-based modeling using ABEL. In *First International Joint Conference on Qualitative and Quantitative Practical Reasoning; ECSQARU–FAPR'97*, pages 171–182, 1997.

[2] S. Benferhat, J. Ben-Naim, R. Jeansoulin, M. Khelfallah, S. Lagrue, O. Papini, N. Wilson, and E. Würbel. Belief revision of gis systems: The results of REV!GIS. In *Proc. Eighth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'05)*, pages 452–464, 2005.

[3] S. Benferhat and H. Prade. Encoding formulas with partially constrained weights in a possibilistic-like many-sorted propositional logic. In *Proc. IJCAI-05*, pages 1281–1286, 2005.

[4] P. Besnard and J. Kohlas. Evidence theory based on general consequence relations. *Int. J. of Foundations of Computer Science*, 6(2):119–135, 1995.

[5] B. Davey and H. Priestley. *Introduction to Lattices and Order (Second edition).* Cambridge University Press, 2002.

[6] J. de Kleer. An assumption-based truth maintenance system. *Artificial Intelligence*, 28:127–162, 1986.

[7] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49:61–95, 1991.

[8] A. P. Dempster. Upper and lower probabilities induced by a multi-valued mapping. *Annals of Mathematical Statistics*, 38:325–39, 1967.

[9] D. Dubois, J. Lang, and H. Prade. Timed possibilistic logic. *Fundamenta Informaticae*, pages 211–234, 1991.

[10] D. Dubois, J. Lang, and H. Prade. Possibilistic logic. In D. Gabbay, C. Hogger, and J. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, pages 439–513. Oxford University Press, 1994.

[11] D. Dubois and H. Prade. *Possibility Theory: An Approach to Computerized Processing and Uncertainty.* Plenum Press, New York, 1988.

[12] D. Dubois and H. Prade. Possibilistic logic: A retrospective and prospective view. *Fuzzy Sets and Systems*, 144(1):3–23, 2004.

[13] R. Haenni, J. Kohlas, and N. Lehmann. Probabilistic argumentation systems. In J. Kohlas and S. Moral, editors, *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, volume 5: Algorithms for Uncertainty and Defeasible Reasoning, pages 221–287. Kluwer, Dordrecht, 2000.

[14] B. Hollunder. An alternative proof method for possibilistic logic and its application to terminological logics. *International Journal of Approximate Reasoning*, 12(2):85–109, 1995.

[15] R. Jeansoulin and E. Würbel. An anytime revision operator for large and uncertain geographic data sets. *Soft Comput*, 7(6):386–393, 2003.

[16] P. Kanjamala, P. Revesz, and Y. Wang. MLPQ/GIS: A GIS using linear constraint databases. In *Proceedings of 9th International Conference On Management Of Data (COMAD'98)*, 1998.

[17] M. Khelfallah and B. Benhamou. Geographic information revision based on constraints. In *Proc. European Conference on Artificial Intelligence (ECAI 2004)*, pages 828–832, 2004.

[18] J. Kohlas. Probabilistic argumentation systems: A new way to combine logic with probability. *Journal of Applied Logic*, 1(3-4):225–253, 2003.

[19] J. Kohlas and P.-A. Monney. Propagating belief functions through constraint systems. *International Journal of Approximate Reasoning*, 5:433–461, 1991.

[20] M. Koubarakis. Tractable disjunctions of linear constraints: Basic results and applications to temporal reasoning. *Theoretical Computer Science*, 266:311–339, 2001.

[21] J. Lang. Possibilistic logic: Complexity and algorithms. In J. Kohlas and S. Moral, editors, *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, volume 5: Algorithms for Uncertainty and Defeasible Reasoning, pages 179–220. Kluwer, Dordrecht, 2000.

[22] S. Moral and N. Wilson. Markov chain Monte-Carlo algorithms for the calculation of Dempster-Shafer belief. In *Proceedings of the Twelfth National Conference on Artificial Intelligence, AAAI-94*, pages 269–274, 1994.

[23] D. Raclot and C. Puech. Photographies aèriennes et inondation: globalisation d'informations floues par un système de contraintes pour définir les niveaux d'eau en zone inondée. *Revue internationale de géomatique*, 8(1):191–206, 1998.

[24] D. Raclot and C. Puech. What does AI contribute to hydrology? aerial photos and flood levels. *Applied Artificial Intelligence*, 17(1):71–86, 2003.

[25] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1, 2):81–132, 1980.

[26] P. Rigaux, M. Scholl, and M. Voisard. *Spatial Databases with Application to GIS*. Morgan Kaufmann, 2001.

[27] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[28] G. Shafer. Constructive probability. *Synthese*, 48:1–60, 1981.

[29] J. Stoer and C. Witzgall. *Convexity and optimization in finite dimensions I*. Springler-Verlag, 1970.

[30] T. Vidal and H. Fargier. Handling contingency in temporal constraint networks: from consistency to controllabilities. *JETAI*, 11:23–45, 1999.

[31] N. Wilson. A Monte-Carlo algorithm for Dempster-Shafer belief. In *Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence*, pages 414–417, 1991.

[32] N. Wilson. Algorithms for Dempster-Shafer theory. In J. Kohlas and S. Moral, editors, *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, volume 5: Algorithms for Uncertainty and Defeasible Reasoning, pages 421–476. Kluwer, Dordrecht, 2000.

[33] N. Wilson. The logic of linear constraints and its application to the flooding problem. Technical report, annex to REVIGIS project year 2, task 1.2 report, 2002.

[34] N. Wilson. Uncertain linear constraints. In *Proc. European Conference on Artificial Intelligence (ECAI 2004)*, pages 231–235, 2004.

[35] N. Wilson. A logic of soft constraints based on partially ordered preferences. *Journal of Heuristics*, 12(4-5):241–262, 2006.

[36] N. Wilson and S. Moral. A logical view of probability. In *Proceedings of the 11th European Conference on Artificial Intelligence, ECAI-94*, pages 386–390, 1994.

[37] E. Würbel, R. Jeansoulin, and O. Papini. Revision: an application in the framework of GIS. In *KR 2000, Principles of Knowledge Representation and Reasoning, Proceedings of the Seventh International Conference*, pages 505–515, 2000.