CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Uncertain linear constraints
Authors	Wilson, Nic
Publication date	2004-08
Original Citation	Wilson, Nic; (2004) 'Uncertain Linear Constraints ', ECAI' 04: Proceedings of the 16th European Conference on Artificial Intelligence, 22- 27 August, Valencia, Spain: IOS Press, pp. 231-235.
Type of publication	Conference item
Link to publisher's version	https://www.iospress.nl/book/ecai-2004/
Rights	© 2004 IOS Press
Download date	2026-11-17 03:47:53
Item downloaded from	https://hdl.handle.net/10468/10763



UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Uncertain Linear Constraints

Nic Wilson[1]

Abstract. Linear constraints occur naturally in many reasoning problems and the information that they represent is often uncertain. There is a difficulty in applying many 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, 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, and a Dempster-Shafer representation.

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. Such constraints can often 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 with a propositional calculus, or sometimes a 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. Therefore it is preferable to represent linear constraints directly, and extend the uncertainty theories to reason with these. We approach this problem by expressing linear constraints in a logic (section 2) and generalising uncertainty formalisms by defining them over this logic. These are illustrated in terms of a simple example. We consider possibilistic logic in section 3, a lattice-valued possibilistic logic and generalised assumption-based reasoning systems in section 4, and Dempster-Shafer theory in section 5.

2 A LOGIC OF LINEAR CONSTRAINTS

In this section we describe 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 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 can also express constraints $a_1x_1 + \cdots + a_nx_n \leq a_0$ and of the form $a_1x_1 + \cdots + a_nx_n = a_0$.

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

The language. Let $V = \{X_1, \ldots, X_n\}$ be a finite set of real-valued variables.[2] 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 $\{1, \ldots, n\}$. $x(i)$, written x_i, is interpreted as a value of the variable 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 $a_1x_1 + \cdots + a_nx_n \geq a_0$.

We are interested in three special constraints $\top$, $\top^0$ and $\bot$. These 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. Linear equations can be represented in the language: $a_1x_1 + \cdots + a_nx_n = a_0$ is equivalent to a pair of constraints $\{a, -a\}$.

We could also easily extend the language to include strict constraints of the form $a_1x_1 + \cdots + a_nx_n > a_0$.

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 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 that satisfy 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:

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

From a and b deduce $a + b$.

For any constraint a, *From $\bot$ 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

[2] 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 [20].

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 [19]) the following result (see [23] for details).

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, for particular sparse systems, Fourier elimination can be efficient, or fast algorithms for special kinds of constraints, such as Simple Temporal Networks [2].

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 [22] 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 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 [14, 15] (see also [25, 9]). An area of land surrounding a flooded river is analysed using aerial photographs and other sources of information. 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 metres above sea level) of compartment i.

We would like to deduce information about the levels x_i for various compartments i. The information we have is of the following form: *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) arising from observations that objects such as vines are partially submerged, and simple linear constraints of the form $x_j \geq x_i$ (call these *rules*), where $j \geq i$, based on observed flows. Both types of information are uncertain, but the flows are less uncertain than the bounds.

This is a special kind of Simple Temporal Problem [2] (though the variables are spatial rather than temporal) for which a simple linear time algorithm can be used involving both upstream and downstream propagations [15, 23], to test consistency and generate inferred bounds on the variables. However, this is not so useful on its own since the input information (owing to its uncertainty) in the application is inconsistent.

3 POSSIBILISTIC LINEAR CONSTRAINTS

In this section it is shown how Possibilistic Logic [7] (Dubois et al., 94) can be extended to deal with linear constraints. In possibility theory [5], degrees of certainty (which are called 'necessity') are assumed to be totally ordered and representable by numbers in $[\,0,1\,]$, with a necessity value of 1, for a proposition, meaning that the proposition is considered completely certain, and a value of 0 meaning no certainty at all. 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.

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 $\mathrm{Nec}_\pi(X) = 1 - \mathrm{Poss}_\pi(\Omega - X)$. 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 $\mathcal{L}$. A possibility distribution on $\mathcal{M}$ induces a possibility measure and a necessity measure on $2^\mathcal{M}$, which induces 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 the pair (a,α), where $a \in \mathcal{L}$ and $\alpha \in [\,0,1\,]$. We assume a set of such pairs $\mathcal{A}$ which can be thought of as an imprecise specification of a necessity measure. Our information is intended to constrain an unknown possibility distribution $\pi : \mathcal{M} \to [\,0,1\,]$. Possibility distribution π is said to satisfy a pair (a,α) if and only its associated necessity measure Nec_π satisfies $\mathrm{Nec}_\pi(a) \geq \alpha$. This is the case if and only if $\pi(x) \leq 1-\alpha$ for all x such that $x \not\models a$. We write in this case that $\pi \models (a,\alpha)$. We say that π satisfies set $\mathcal{A}$ of such pairs if and only if π satisfies each of the pairs in $\mathcal{A}$. 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 $\pi \models \mathcal{A}$.

For set of pairs $\mathcal{A}$ and $\beta \in [\,0,1\,]$ define $\mathcal{A}_\beta$ to be $\{a : (a,\alpha) \in \mathcal{A},\ \alpha \geq \beta\}$. We have the following key result for possibilistic logic of linear constraints, which connects entailment in the possibilistic logic with entailment in the linear constraints logic.

Theorem 2 *For set of pairs $\mathcal{A}$, we have $\mathcal{A} \models (b,\beta)$ if and only if for all $\gamma < \beta$, $\mathcal{A}_\gamma \models b$. Let $b^* = \sup\{\gamma : \mathcal{A}_\gamma \models b\}$. Then $\mathcal{A} \models (b,b^*)$, and if $\mathcal{A} \models (b,\beta)$ then $\beta \leq b^*$, so that b^* is the strongest necessity value deducible, given $\mathcal{A}$, for b.*

For finite $\mathcal{A}$ we have $\mathcal{A} \models (b,\beta) \iff \mathcal{A}_\beta \models b$.

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

The sets of pairs $\mathcal{A}$ can be used to represent a possibilistic linear constraints knowledge base. If we are interested in finding information about the certainty (necessity) degree of b then we can use, for example, a binary search over $[\,0,1\,]$ to find increasingly large values β 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 linear constraints language (so computational efficiency of the procedure is closely tied to the efficiency of deduction in the class of linear constraints used). Of particular interest is if we can find $\beta > \bot^*$ with $\mathcal{A} \models (b,\beta)$, as this indicates positive support for b. This can be also used for a possibilistic approach to belief revision.

Possibilistic constraints for the flooded river example

We can assign necessity values to the various bounds and rules in the flooding problem (and similarly, for other Simple Temporal Problems [2]). 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 rule, a lower bound or an upper bound.

For any α appearing in some pair 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. Applying this approach for each α will then give us the (maximum) 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 rule* $x_j \geq x_i$ *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 rule 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 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 $\bot^*$ 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.

4 LATTICE-VALUED POSSIBILISTIC LOGIC

Many results on possibilistic logic can be generalised to a situation where the values of necessity are in a distributive lattice [6, 7]. Let $\mathcal{K} = (K, 0, 1, \bigwedge, \bigvee)$ be a completely distributive lattice [4], with greatest lower bound operation $\bigwedge$ and least upper bound operation $\bigvee$ on subsets of K satisfying infinite distributive properties. 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 $\mathcal{P}$ to consist of all pairs (A, α) where $A \subseteq \mathcal{L}$ is a set of linear constraints,[3] and $\alpha \in K$ is a lattice element. Set A of linear constraints 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. The definition in the standard case is equivalent to $\pi \models (A, \alpha)$ 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 $(\alpha^*)^* = \alpha$. Models 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 $x \not\models A$, $(\pi(x))^* \succeq \alpha$. 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, because of the potentially more complex structure of the lattice (in particular it being generally only partially ordered). But we can still define a 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* $(\bigcup_{i \in I} A_i, \bigwedge_{i \in I} \alpha_i)$.

Define the set of syntactic consequences $\mathcal{C}(\Delta)$ of $\Delta \subseteq \mathcal{P}$ to be the the intersection of all sets $\Gamma \subseteq \mathcal{P}$ (which is the unique smallest set Γ) such that $\Gamma \supseteq \Delta$ and 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.

Theorem 3 (Soundness and Completeness of Paired System) $\Delta \models (B, \beta)$ *if and only if* $\Delta \vdash (B, \beta)$, *where* $\Delta \cup \{(B, \beta)\} \subseteq \mathcal{P}$.

As usual, soundness is easy to confirm. We sketch how completeness is proved. Label Δ as $\{(A_i, \alpha_i) : i \in I\}$, and define, for $x \in \mathcal{M}$, $I_x = \{i \in I : x \not\models A_i\}$. Define model π_Δ by, for $x \in \mathcal{M}$, $(\pi_\Delta(x))^* = \bigvee_{i \in I_x} \alpha_i$. It can be easily shown that π_Δ satisfies Δ. Suppose $\Delta \models (B, \beta)$; then π_Δ satisfies (B, β), which leads to $\bigwedge_{x \not\models B} \bigvee_{i \in I_x} \alpha_i \succeq \beta$.

Let $S_B = \{\sigma \subseteq I : \bigcup_{i \in \sigma} A_i \models B\}$. Applying the third inference rule, then the first inference rule, then the second one leads to: $\mathcal{C}(\Delta)$ contains the pair $\left(B, \bigvee_{\sigma \in S_B} \bigwedge_{i \in \sigma} \alpha_i\right)$. The distributivity property can be used to show that $\bigwedge_{x \not\models B} \bigvee_{i \in I_x} \alpha_i$ equals $\bigvee_{\sigma \in S_B} \bigwedge_{i \in \sigma} \alpha_i$, which implies that $\mathcal{C}(\Delta)$ contains (B, β) using the first inference rule. So $\Delta \vdash (B, \beta)$ as required.

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. 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 A^α is the union of A over all $(A, \alpha) \in \Delta$.

Assumption-based reasoning

The above soundness and completeness results can be applied to give similar results for argumentation systems which may be viewed as generalised versions of Assumption-Based Truth Maintenance Systems [10]. For example, consider a system of pairs (A, ϕ) where A is a set of linear constraints, and ϕ is a formula in some propositional language $\mathcal{R}$; ϕ is intended to represent conditions under which constraints A are known to hold. To express relationships between these conditions, it can be useful also to allow an additional set of formulae $T \subseteq \mathcal{R}$.

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

[3] The results in this section (in particular, theorems 3 and 4, as well as Theorem 2 in the last section) do not depend at all on the internal structure of the language $\mathcal{L}$ or of the set of models $\mathcal{M}$; the same results hold given arbitrary sets $\mathcal{L}$ and $\mathcal{M}$ with arbitrary relation $\models \subseteq \mathcal{M} \times \mathcal{L}$ used to define the semantic entailment relation between subsets of $\mathcal{L}$. In particular the results hold if the language $\mathcal{L}$ of linear constraints is extended with disjunctions and/or conjunctions and/or negations.

possible assignment to both the real-valued variables and the propositional variables. (A, ϕ) restricts possible models (x, ω). Pair (A, ϕ) represents that, if condition ϕ holds, then all of A hold; so we say (x, ω) satisfies (A, ϕ) if [ω satisfies ϕ implies $x \models A$]. Therefore (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* $(A \cup B, \phi \wedge \psi)$.

Theorem 4 *With the above proof theory, finite Δ syntactically entails (B, ψ) if and only if Δ semantically entails (B, ψ).*

Given Δ one can associate with a $B \subseteq \mathcal{L}$ 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 $T \cup \{\phi\} \models \phi_B$. 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 proved, 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 independent and has a chance r_i of holding, this generates a probability measure on $\mathcal{R}$ and hence degrees of belief. This situation is also a special case of the generalised Dempster-Shafer theory described in the next section (cf., work on probabilistic argumentation systems [8, 1]).

5 DEMPSTER-SHAFER ON LINEAR CONSTRAINTS

This section shows how Dempster-Shafer theory can be extended to reason with linear constraints. See also [11] for a related approach to Dempster-Shafer for spatial and temporal reasoning.

The formalism of (Shafer, 76) [17] was derived from that of Arthur Dempster [3]; Dempster's framework is more convenient for our purposes, and we describe a slight variant of it. 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 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 $S = (\Omega, \mathrm{P}, \Gamma)$ 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: $\mathrm{Bel}_S(A) = \mathrm{P}(\Gamma(\omega) \models A)$. Belief functions are intended as representations of subjective degrees of belief, as described in (Shafer 76; 81) [17, 18].

Dempster's rule of combination. 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 combined effect of these, given the appropriate independence assumptions, is calculated using Dempster's rule. The combination $(\Omega, \mathrm{P}_{\mathrm{DS}}, \Gamma)$ of these source triples over $\mathcal{L}$ 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 $\Gamma' \colon \Omega^\times \to \mathcal{F}$ 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 $\mathrm{Bel}(A) = \mathrm{P}_{\mathrm{DS}}(\Gamma(\omega) \models A)$, which equals $\mathrm{P}_{\mathrm{DS}}(\Gamma(\omega) \vdash A)$ since $\Gamma(\omega)$ is a finite subset of $\mathcal{L}$. Alternatively, we could map each source triple to its corresponding source triple over $\mathcal{M}$, combine the source triples and generate function Bel_0 over $\mathcal{M}$. We then have $\mathrm{Bel}(A) = \mathrm{Bel}_0([A])$.

Computing combined belief. It is possible to adapt various of the standard approaches (see e.g., [21]) for computing combined belief to the uncertain linear constraints scenarios. In particular, various Monte-Carlo algorithms can be adapted to give arbitrarily close approximations of values of belief.

Since, for finite $A \subseteq \mathcal{L}$, $\mathrm{Bel}(A) = \mathrm{P}_{\mathrm{DS}}(\Gamma(\omega) \vdash A)$, 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 $\mathrm{P}_{\mathrm{DS}}(\omega)$ and say that the trial succeeds if $\Gamma(\omega) \vdash A$, 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 $\mathrm{P}_{\mathrm{DS}}(\omega)$ 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 is bounded, this algorithm has low complexity, proportional to the complexity of proof in the logic [21], 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 [13, 21].

Dempster-Shafer for the flooded river example

An uncertain rule $x_j \geq x_i$ with reliability $p \in [0, 1]$, might be represented as a source triple $(\{\omega, \omega'\}, \mathrm{P}, \Gamma)$, with $\mathrm{P}(\omega) = p$, $\mathrm{P}(\omega') = 1 - p$ and $\Gamma(\omega)$ being $\{x_j \geq x_i\}$, and $\Gamma(\omega') = \{\top\}$ (or alternatively the empty set of constraints). This corresponds to a *simple support function* [17]. Given just this source triple, we can deduce the rule $x_j \geq x_i$ with chance p (and with chance $1 - p$ we deduce nothing).

Suppose, for example, our information makes us absolutely certain that the level of compartment 1 is at least 75, and strongly suggests it is at least 95, with two more tentative lower bounds of 110 and 120. We might use source triple $(\Omega_1, \mathrm{P}_1, \Gamma_1)$ where $\Omega_1 = \{1, 2, 3, 4\}$, $\mathrm{P}_1(1) = 0.1$, $\mathrm{P}_1(2) = 0.1$, $\mathrm{P}_1(3) = 0.4$, and $\mathrm{P}_1(4) = 0.4$; $\Gamma_1(1)$ is the constraint $x_1 \geq 120$, $\Gamma_1(2) = (x_1 \geq 110)$, $\Gamma_1(3) = (x_1 \geq 95)$, and $\Gamma_1(4) = (x_1 \geq 75)$. This source triple can be thought of as a constraint $x_1 \geq l_1$ where l_1 is a random variable, taking values 120, 110, 95 and 75 with chances 0.1, 0.1, 0.4 and 0.4, respectively.

In this way we can model the uncertain upper and lower bounds for the compartments. Given that the appropriate independence as-

sumptions are satisfied,[4] then we can use Dempster's rule to combine the information. 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 $x_6 \in [120, 130]$. We can find which compartments have tightly constrained bounds, and for which our information is poorer. At the same time we can find smaller intervals containing x_i, associated with a smaller degree of belief, but which more tightly constrain the variable and enable us to make an educated guess at the water level.

6 DISCUSSION

This paper shows how a number of the most important uncertainty formalisms can be extended to deal with uncertain linear constraints. The formalisms we discuss are possibilistic logic, a generalised form of possibilistic logic which encompasses a general form of assumption-based reasoning, and Dempster-Shafer theory. Our approach, based on a logical representation of linear constraints, can also be easily applied to other uncertainty formalisms, such as some non-monotonic logics and belief revision formalisms [24].

To simplify the presentation, we presented a rather basic language of linear constraints in section 2. Allowing, for example, disjunctions of linear constraints (see e.g., [2, 12]) is important for many applications, for example, job shop scheduling. The formalisms for reasoning with uncertain linear constraints described in this paper generalise immediately to more complex logical languages such as these (see, in particular, footnote 3).

ACKNOWLEDGEMENTS

This work was supported by the REVIGIS project, IST-1999-14189, and by Science Foundation Ireland under Grant 00/PI.1/C075. It benefitted from 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. I am also very grateful to Angela Clutterbuck, without whom this submission would not have been possible.

REFERENCES

[1] Anrig, B., Haenni, R., Kohlas, J., and Lehmann, N. Assumption-based Modeling using ABEL. D. Gabbay, R. Kruse, A. Nonnengart and H.J. Ohlbach (eds.), *First International Joint Conference on Qualitative and Quantitative Practical Reasoning; ECSQARU–FAPR'97*. Springer. 1997.

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

[3] Dempster, A. P., Upper and Lower Probabilities Induced by a Multivalued Mapping. *Annals of Mathematical Statistics* 38: 325–39 (1967).

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

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

[6] Dubois, D., Lang, J., Prade, H., Timed Possibilistic Logic, *Fundamenta Informaticae*, XV: 211–234 (1991).

[7] Dubois, D., Lang, J., Prade, H., Possibilistic Logic, *Handbook of Logic in Artificial Intelligence and Logic Programming*, D. Gabbay, C. Hogger, J. Robinson (eds.), Vol. 3, 439–513, Oxford University Press (1994).

[4] It is also possible to model dependencies in the information, by constructing a different combined probability function to P_{DS}.

[8] Haenni, R., Kohlas, J., and Lehmann, N.. Probabilistic Argumentation Systems. Pages 221–287 of: J. Kohlas and S. Moral (eds.), *Handbook of Defeasible Reasoning and Uncertainty Management Systems, Volume 5: Algorithms for Uncertainty and Defeasible Reasoning*. Kluwer, Dordrecht. 2000.

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

[10] de Kleer, J., An Assumption-based Truth Maintenance System, *Artificial Intelligence Journal* 28: 127–162 (1986).

[11] Kohlas, J., and Monney, P.A. Propagating Belief Functions Through Constraint Systems. *Int. J. Approximate Reasoning*, 5, 433-461. (1991).

[12] Koubarakis, M., Tractable Disjunctions of Linear Constraints: Basic Results and Applications to Temporal Reasoning, *Theoretical Computer Science*, Vol. 266, pages 311–339, September 2001.

[13] Moral, S., and Wilson, N., Markov Chain Monte-Carlo Algorithms for the Calculation of Dempster-Shafer Belief, *Proceedings of the Twelfth National Conference on Artificial Intelligence, AAAI-94*, Seattle, USA, July 31–August 4, 1994, 269–274, (1994).

[14] Raclot, D., Puech, C.: 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).

[15] Raclot, D., and Puech, C.: What Does AI Contribute to Hydrology? Aerial Photos and Flood Levels. *Applied Artificial Intelligence* 17(1): 71-86 (2003).

[16] Reiter, R., A Logic for Default Reasoning, *Artificial Intelligence Journal* 13 (Nos. 1, 2): 81–132 (1980).

[17] Shafer, G., 76, *A Mathematical Theory of Evidence*, Princeton University Press, Princeton, NJ (1976).

[18] Shafer, G., Constructive Probability, *Synthese*, 48: 1–60 (1981).

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

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

[21] Wilson, N., Algorithms for Dempster-Shafer Theory, in Kohlas, J., and Moral, S., (eds.), *Volume 5: Algorithms for Uncertainty and Defeasible Reasoning*, *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, (Series eds.: D. Gabbay, P. Smets), Kluwer Academic Publishers (2000).

[22] Wilson, N., and Moral, Serafín, 94, A Logical View of Probability, *Proceedings of the 11th European Conference on Artificial Intelligence, ECAI-94)*, Amsterdam, The Netherlands, August 8–12,1994, 386–390 (1994).

[23] Wilson, N., The logic of linear constraints and its application to the flooding Problem, annex to REVIGIS project year 2, task 1.2 report. (2002).

[24] Wilson, N., Uncertain linear constraints and their application to the flooding problem, annex to REVIGIS project year 3, task 1.2 report (2003).

[25] Würbel, E., Jeansoulin, R., Papini, O. Revision: an application in the framework of GIS, In proc. of KR'00 (2000).