Geographic information revision based on constraints

Mahat Khelfallah and **Belaïd Benhamou**[1]

Abstract. *Information usually has many sources and is often incomplete and / or uncertain, this leads to many inconsistencies. Revision is the operation which consists in identifying these inconsistencies and then in removing them by changing a minimum of information. In this paper, we are interested in geographic information revision in the framework of a flooding problem. We show how to express and how to revise this problem by using simple linear constraints. We present three revision strategies based on linear constraints resolution. We apply and compare these approaches on both a real-world flooding problem and random flooding instances.*

Keywords

Revision, Linear constraints, Geographic information.

1 Introduction

Many research works have been done in the field of knowledge revision [1, 3, 2]. Revision is the operation which consists in restoring the consistency of a knowledge base by considering other more reliable information. It identifies the inconsistencies, then corrects them by keeping a maximum of the initial information unchanged.

Some geographic information can be expressed by spatial and / or temporal constraints which sometimes are linear inequalities [9, 4]. These constraints have good complexity properties which can be exploited in the revision process.

In this paper, we are interested in geographic knowledge revision based on linear constraints resolution in the framework of a flooding problem. We propose three different revision strategies:
- *The all conflicts revision strategy* which first, identifies all the conflicts of the problem, and then computes a smallest subset of constraints whose revision is sufficient to restore the consistency.
- *The conflict by conflict revision strategy* which revises one by one the detected conflicts. A correction is made as soon as a conflict is detected.
- *The hybrid revision strategy* which identifies a representative part of the different conflicts, then corrects them in an optimal way with respect to the number of revised constraints. This operation is repeated till elimination of all the conflicts.

The rest of this paper is organized as follows. In section 2, we recall some background on linear constraints. We describe in section 3 the flooding problem and show how it can be represented by linear constraints. Revision principle is presented in section 4. Section 5 defines three revision approaches, namely, the *all conflicts revision*, the *conflict by conflict revision*, and the *hybrid revision*. We experiment in section 6 the different revision methods on a real flooding application and on random flooding instances, and make a comparison on the obtained results. Section 7 compares our work with previous related works on the flooding problem. Section 8 concludes the work.

2 Background

A *linear constraint network* N is defined by N=(X, C) where X is a finite set of variables $X_0, ..., X_n$, having continuous domains. C is the set of constraints defined on these variables. Each constraint of C is of the form $X_j - X_i \leq a_{ij}$.

A tuple $X = (x_0, ..., x_n)$ of real values is a solution of the linear constraint network N if the instantiation $\{X_1$=$x_0, \ldots, X_n$=$x_n\}$ satisfies all its constraints. The linear constraint network N is *consistent* if only if it has a solution.

We associate to the linear constraint network N=(X, C) a directed edge-weighted graph, $G_d = (X, E_d)$, called a *distance graph*. X is the set of vertices corresponding to the variables of the network N, and E_d is the set of arcs representing the set of constraints C. Each constraint X_j - $X_i \leq a_{ij}$ in C is represented by the arc $i \rightarrow j$ [2], which is weighted by a_{ij}.

Now, we are able to describe and represent the flooding problem which we use to explain our revision approaches.

3 The Flooding problem

3.1 Description of the flooding problem

During a flooding in the Hérault valley (in the south of France), a part of this area was studied in order to get correct estimates of the water heights (above the sea level) from aerial photographs and topographic knowledge of the region. The studied part was divided into n parcels, which are enough small to consider the water height within each parcel constant.

Two sources of information are available. The first one concerns estimates of water height in each parcel. Each estimate is given as an interval of possible values and is represented by its upper and lower bounds.

The second source represents observations on hydraulic relations between some adjacent parcels. These relations can be either flow or balance relations. A flow from the parcel i to the parcel j is observed when the water height in the parcel i is greater than the one in the parcel j. A hydraulic balance between two parcels expresses equality between their corresponding water heights. For a more detailed description, see [7, 8].

Each of the two considered sources of information is consistent separately, but conflicts appear when both sources are merged. Moreover, the source related to observations on hydraulic relations is more reliable than the source related to water heights estimates. The water height estimates are *revised* by the hydraulic relations.

[1] LSIS - Provence university at Marseille, France email: {mahat,benhamou}@cmi.univ-mrs.fr

[2] For simplicity vertices of the graph G_d are denoted by the indices of the variable of X.

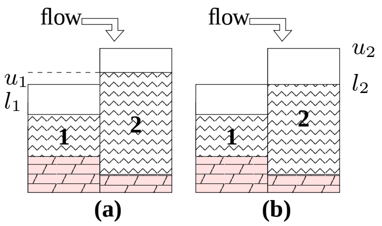


Figure 1. An example of a conflict between the estimations of the water heights and the hydraulic relations

Figure 1 illustrates an example of a conflict between the two sources of information. Schema (a) shows a conflict between the water height in the two parcels and the observed flow from the parcel 1 to the parcel 2. The lower bound of water height in parcel 2 has been revised in schema (b), and the conflict appearing in (a) is removed.

We will see in the sequel how to detect such conflicts and how to correct them. Before doing this, we give a representation of the flooding problem in the linear constraint network framework.

3.2 Representation of the problem

The flooding problem is represented by the linear constraint network $N = (X, C)$ where X is the set of variables X_i corresponding to the water height in parcel i $(i \neq 0)$, and $X_0 = 0$ represents the sea level. C is the set of constraints representing both estimates on water heights and the hydraulic relations. The constraints are defined as follows:
Estimates on water heights : Each parcel i is associated with the constraints $l_i \leq X_i - X_0$ and $X_i - X_0 \leq u_i$, where l_i and u_i are respectively the lower and the upper bounds of the interval delimiting the water height in the parcel i.
Observations on hydraulic relations : Each observed flow from parcel i to parcel j is expressed by the constraint $X_j - X_i \leq 0$; and each observed balance between parcels i and j is represented by the constraints: $X_j - X_i \leq 0$ and $X_i - X_j \leq 0$.

4 Revision

The revision principle we adopt consists in identifying the conflicts of the problem, and changing a minimal number of constraints to restore the consistency.

4.1 Detection of conflicts

We present a method which detects inconsistencies in a set of constraints. For this, we use an important result which stipulates that *a linear constraint network is consistent if and only if its corresponding distance graph does not contain negative circuits* [3] [10, 6, 5].

To detect the negative circuits in a distance graph, we use the result of the following proposition.

Proposition 1 *An elementary circuit in the distance graph corresponding to a flooding problem is negative if only if it includes a path $\{i, 0, j\}$ whose distance is negative.*

Proof 1 *Suppose that the distance graph G_d of a given flooding constraint network contains an elementary negative circuit C. The circuit C must include two arcs $(i, 0)$ and $(0, j)$ incident to the origin 0 and for which the sum of the corresponding labels is negative. Otherwise, the circuit C cannot be negative, since all the labels of the others arcs non incident to the origin are equal to zero. Therefore, C includes a negative path $\{i, 0, j\}$. Conversely, each elementary circuit of the distance graph including a negative path $\{i, 0, j\}$ must be negative, since all the other arcs are weighted by zero.*

The previous proposition states that each conflict is due to an elementary negative circuit including a negative path $\{i, 0, j\}$. To find all the conflicts, one has to enumerate all the pairs (i, j) participating in the different negative paths $\{i, 0, j\}$ and check existence of an elementary path from j to i which does not visit 0. The procedure of detection of all the conflicts is given in the figure 2. The function $Distance_{G_d}(p)$ returns the sum of the weights of the arcs forming the path p of G_d. $Path_{G_d}(j, i)$ checks if there exists a path from j to i in G_d [4].

procedure Detection-Conflict(G_d: the distance graph)
begin
  For $i, j = 1\ to\ n\ (i \neq j)$
  begin
    If $Distance_{G_d}(\{i, 0, j\}) < 0$ **and** $Path_{G_d}(j, i)$ **then**
    Conflict between $l_i \leq X_i$ - X_0 and X_j - $X_0 \leq u_j$
  end
end

Figure 2. Detection of conflicts algorithm

Let n be the number of vertices of the distance graph G_d and e the number of its arcs. The previous procedure performs n^2 iterations. Both functions $Distance_{G_d}$ and $Path_{G_d}$ are called in each iteration. The function $Distance_{G_d}$ is elementary and has a constant time complexity ($O(2)$). The time complexity of the function $Path_{G_d}$ is in $O(e)$. Thus, the time complexity of the procedure is $O(n^2 e)$ in the worst case.

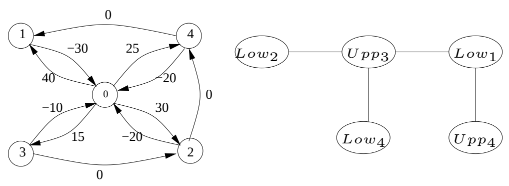
4.2 Representation of the conflicts

We recall that we are interested in revising estimates on water heights in the parcels by the hydraulic relations existing between these parcels. The conflicting constraints are those corresponding to some pairs of arcs incident to the vertex 0, for instance the arc $0 \longrightarrow j$ (respectively the arc $i \rightarrow 0$) represented by the constraint X_j - $X_0 \leq u_j$ (respectively the constraint X_0 - $X_i \leq$ - l_i).

Each conflict of the problem is defined between an upper bound of a variable X_j and a lower bound of a variable X_i. If we consider the set of vertices V=$\{Low_1, Upp_1, \ldots, Low_i, Upp_i, \ldots, Low_n, Upp_n\}$ corresponding to the lower and upper bounds (one vertex per bound) of water heights in the conflicting parcels, then every conflict between the lower bound of X_i and the upper bound of X_j is expressed by the edge (Low_i, Upp_j). If E is the set of all these edges, then the set of conflicts is represented by the graph G_c = (V, E) which we call *the graph of conflicts*.

[3] A negative circuit is a circuit whose the sum of its arcs labels is negative.

[4] These functions will be used also in the procedures of the next section.



a. Graph of distances **b. Graph of conflicts**

Figure 3. The graph of distances and the graph of conflicts

Example 1 *We consider a set of four parcels whose the water heights are represented by the set of variables $\{X_1, X_2, X_3, X_4\}$. The water height estimates and the observed hydraulic relations are represented by the distance graph in the figure 3.a. From the negative circuit [0 , 3 , 2 , 4 , 1 , 0], we deduce the conflicting constraints X_3 - $X_0 \leq 15$ and X_0 - $X_1 \leq -30$. In other words, the upper bound of the water height of the parcel 3 is in contradiction with the lower bound of the water height of the parcel 1. This adds the edge (Upp_3, Low_1) in the conflict graph. By considering all the conflicting constraints appearing in all the elementary negative circuits, we obtain the conflict graph of the figure 3.b.*

4.3 Computing the smallest subsets of constraints to revise

To remove all the detected conflicts, some constraints involved in them have to be revised. To respect minimal change principle, we have to look for the smallest subsets of constraints whose the revision is sufficient to restore the consistency. This operation amounts to looking for the minimal coverings of the conflict graph. We recall the definition of a covering and a minimal covering of a graph.

Definition 1 *Let G be a graph defined by $G = (V, E)$. T is a* covering *of the graph G if and only if T is a subset of the set of vertices V, and for each edge (x, y) in E, $T \cap \{x, y\} \neq \emptyset$.*

Definition 2 *A covering T of a graph G is minimal if and only if for each covering T' of G, if $\mid T' \mid \leq \mid T \mid$ then $|T'| = |T|$.*

Example 2 *The conflict graph of the figure 3.b has two minimal coverings: T_1=$\{Upp_3, Low_1\}$ and T_2=$\{Upp_3, Upp_4\}$. The correction of the constraints involving Upp_3 and Low_1 (respectively, involving Upp_3 and Upp_4) is sufficient to remove all the conflicts.*

The minimal covering problem is known to be a NP-hard problem. Our method of minimal covering search is a backtracking algorithm defined on the vertices of the graph of conflicts. This algorithm does not enumerate systematically all the coverings of the graph of conflicts to find the minimal ones. Indeed, it does not explore the branches of the search tree leading to partial coverings having greater cardinalities than the one of the current minimal coverings. To reduce the minimal covering search space, a heuristic which selects the vertex having the highest degree is used.

Generally, we obtain more than one minimal covering. We need information of an expert to choose one among the obtained coverings. For instance, an expert may prefer revising constraints related to downstream parcels than constraints related to upstream parcels. However, in absence of further information, we can either consider all the obtained coverings or choose arbitrary one among them.

4.4 Correction of the constraints

We shall see now, how to perform the corrections. Let be a conflict between the two constraints $X_j - X_0 \leq u_j$ and $l_i \leq X_i - X_0$. Elimination of this conflict needs the correction of either the upper bound of X_j, or the correction of the lower bound of X_i [5].

If d_{ii} is the distance of the negative circuit which identifies the previous conflict, then replacing the constraint $X_j - X_0 \leq u_j$ (respectively, the constraint $l_i \leq X_i - X_0$) by the constraint $X_j - X_0 \leq u_j$ - d_{ii} (respectively, the constraint l_i + $d_{ii} \leq X_i - X_0$) will correct the conflict and will eliminate the corresponding negative circuits.

Example 3 *Consider example 1. A conflict between the constraints X_3 - $X_0 \leq$ 15 and X_0 - $X_1 \leq$ -30 is detected. The distance of the corresponding negative circuit is -15. This conflict can be corrected either by increasing the upper bound of X_3 by 15, or by decreasing the lower bound of X_1 by 15. If we choose the first alternative, we will replace the constraint X_3 - $X_0 \leq$ 15 by the constraint X_3 - X_0 $\leq$ 30.*

Proposition 2 *The correction of a constraint cannot cause apparition of new conflicts.*

Proof 2 *The proof is trivial since the modification of the involved bound makes the interval of the corresponding variable larger. It is just a constraint weakening.*

We present now the three revision strategies we propose.

5 Revision strategies

5.1 The all conflicts revision

This approach detects all the conflicts of the problem, then corrects them by weakening a minimal number of conflicting constraints. The *all conflicts revision* algorithm is sketched in figure 4.

procedure All-conflicts(G_d: the distance graph)
begin
  $Detection\text{-}Conflict(G_d)$
  Construct G_c the graph of conflicts
  Compute minimal coverings of G_c
  Choose one minimal covering [6]and correct its corresponding constraints
end

Figure 4. All conflicts revision algorithm

To evaluate the complexity of this algorithm, we proceed step by step. Let n be the number of vertices in the distance graph and e the number of its arcs. The detection of all the conflicts is in $O(n^2e)$. Construction of the conflict graph is in $O(n^2)$, since the number of conflicts is at most $n(n-1)$. The minimal covering algorithm is a backtracking procedure whose complexity is $O(2^{2n})$ where $2n$ is the maximal number of vertices that can contain the graph of conflicts G_c. The correction of constraints is in $O(n)$. Therefore, the complexity of the All conflicts algorithm is $O(4^n)$ in the worst case.

[5] The correction of the both bounds is not considered because of the strategy of minimizing the number of revised constraints.

[6] As pointed out above, we can consider all the minimal coverings. In this case, we have to correct for each one its corresponding constraints.

5.2 The conflict by conflict revision

An intuitive idea to restore the consistency is to correct a conflict whenever it is detected. This implies revising the corresponding conflicting constraints as soon as they are identified. The procedure of this method is sketched in figure 5.

procedure Conflict-Conflict(G_d: the distance graph)
begin
  For $i, j = 1\, to\, n\ (i \neq j)$
  begin
    If $Distance_{G_d}(\{i, 0, j\}) < 0$ **and** $Path_{G_d}(j, i)$ **then**
    Correct one of the constraints $l_i \leq X_i - X_0$ or $X_j - X_0 \leq u_j$
  end
end

Figure 5. Conflict by conflict revision algorithm

This algorithm performs n^2 iterations. Each iteration is in $O(e)$ in the worst case. The complexity of the conflict by conflict revision algorithm is $O(n^2e)$ in the worst case. This algorithm has a good complexity, but it does not respect minimality criterion of revision. A third method which is a hybrid of both previous ones, is presented in the following.

5.3 The Hybrid revision

This approach consists first in identifying at each iteration a representative subset of conflicts. That is, a negative circuit containing each vertex participating to some conflicts is computed. The second step consists in revising the constraints corresponding to these conflicts. Both steps are repeated until the restoration of consistency.

The subset of conflicts of each iteration is given by running the *Detection-conflict-Hybrid* procedure on the current distance graph. This procedure is a slight modified version of the *Detection-Conflict* procedure of figure 2. It detects at most one negative circuit for each vertex involved in a conflict. Thus, it computes at most n conflicts corresponding to the representative subset of conflicts which we shall revise. Revision by this method is relatively minimal to the subset of conflicts treated at each iteration. The procedure of this method is sketched in figure 6.

procedure hybrid(G_d: the distance graph)
begin
  repeat
  $Detection\text{-}Conflict\text{-}Hybrid(G_d)$
  Construct G_c the graph of conflicts
  Compute a minimal covering of G_c
  Correct the constraints corresponding to the minimal covering
  until consistency is restored
end

Figure 6. The hybrid algorithm

This algorithm terminates because the revision of detected negative circuits cannot generate new ones and the number of conflicts in the initial distance graph is equal at most to $n(n-1)$. Therefore, the algorithm performs a number of iterations which is bounded by $(n-1)$ since n conflicts are handled in each iteration.

To evaluate the complexity of each iteration, let g_i be the number of the detected conflicts at the iteration i, g_i is bounded by n. The complexity of the *Detection-Conflict-Hybrid* procedure is $O(n^2e)$ in the worst case. Construction of the graph of conflicts of an iteration i is in $O(g_i)$. Computing a minimal covering of the graph of conflicts corresponding to the iteration i is in $O(2^{2g_i})$. The correction of the conflicts of the iteration i is in $O(g_i)$. Thus, an iteration of the hybrid revision algorithm is in $O(2^{2g_i})$ and is $O(2^{2n})$ in the worst case, and the total complexity of this algorithm is $O(n2^{2n})$ in the worst case. In practice, the number of iteration does never reach the worst case, since correction of a detected conflict can eliminate other not yet detected conflicts.

6 Experimental results

The three revision algorithm presented in this paper are implemented in C and run on a pentium 4, 2.4 MHz with 512 MB of RAM. They are tested on the real-world flooding problem in the Hérault valley and random flooding instances.

6.1 The real-world flooding problem

This problem contains 180 parcels. To each of these 180 parcels is associated a variable which represents its water height and two constraints which represent the interval of possible values for this height. There are 360 constraints of water height estimates. Furthermore, there are about 270 constraints representing the flow and balance relationships between parcels. The problem has 180 variables and 630 constraints. Application of the three revision algorithms to this problem gives the results of table 1.

Revision methods	All conflicts	Conf. by conf.	Hybrid
# detected conflicts	15	11	13
# corrections	10	11	10
Time (s)	0	0	2

Table 1. Results on the real flooding problem

We can see that the conflict by conflict revision algorithm has performed more corrections than the two other approaches, because it does not minimize the corrections. The conflict by conflict revision algorithm detects less conflicts than the other algorithms. This is due to the propagation of some corrections which eliminate not yet detected conflicts. Both the All-conflicts and the Hybrid methods have similar results on this problem.

The real flooding application is simple and cannot provide complete arguments for comparing the three methods. We then experiment them on random instances of the flooding problem.

6.2 Random flooding instances

Generation of random flooding problems is based on two parameters: n the number of variables, and d the constraint density which is a ratio of the number of constraints to the number of possible constraints. Table 2 shows some of the obtained results.

We can see that when the density grows, the number of detected conflicts, and the number of corrections increase for all the methods.

Revision		All conflicts			Conflict by conflict			Hybrid		
		Density			Density			Density		
		0.2	0.5	0.8	0.2	0.5	0.8	0.2	0.5	0.8
n=100	# conflicts	2188	2243	2299	120	127	135	1794	1685	1724
	# corrections	52	53	55	120	127	135	92	76	78
	Time (s)	48	52	54	0	0	0	0	0	0
n=500	# conflicts	-	-	-	573	598	633	44937	46201	45798
	# corrections	-	-	-	573	598	633	498	483	468
	Time (s)	-	-	-	0	0	0	7	6	5
n=1000	# conflicts	-	-	-	1167	1200	1232	181748	183988	183612
	# corrections	-	-	-	1167	1200	1232	952	959	1044
	Time (s)	-	-	-	0	0	0	75	64	57

Table 2. Results on the revision of random flooding instances

The All conflicts method corrects a minimal number of constraints and the hybrid method corrects less constraints than the Conflict by conflict method. The All conflict method reaches its limit when the number of variables approximates 150. The two other methods are able to solve greater variable size problems as shows table 2.

7 Related work

Würbel et al. [11, 12] have proposed three methods for geographic information revision in the context of the flooding application. They represent the flooding problem by propositional clauses and compute the smallest subsets of clauses whose the removal restores the consistency. That is what they call *remove sets*. Assuming the same definition of a conflict, the removed sets computed in [11, 12] and the smallest subsets of constraints to revise obtained in our *All conflicts revision* concern the same estimates on water heights. However, Würbel et al.'s methods can not revise the entire flooding problem. Their best results are obtained with the REMr method which revises only a part of 12 parcels, while the *All conflicts revision* deals with the entire application with its 180 parcels in less than one second. Our method outperforms significantly the REMr algorithm.

Another work on the flooding application was done by Raclot and Puech [7, 8]. They propose a method which consists in detecting conflicts by propagating constraints on lower and upper bounds of the water heights intervals. Like in our *Conflict by conflict revision*, a conflict is corrected as soon as it is detected. However, their method is limited to acyclic hydraulic relations, and does not handle hydraulic balances as in our methods.

8 Conclusion

We have presented in this paper three approaches of revision. These methods adopt different strategies. The first one computes all the conflicts and correct them with respect to the minimal change principle. Because of the high complexity of the minimal covering search, the performance of this method decreases when the problem size grows. One way to increase its efficiency is to consider just a good covering rather than a minimal one. The second strategy proceeds conflict per conflict. It revises a conflict as soon as it is detected. This approach could be efficient for applications with a great number of inter-dependant conflicts. However it does not respect the minimal change criterion. The third method is a hybrid of the two former ones. It detects a subset of conflicts, corrects them, and repeats this process until the restoration of the consistency. It constitutes a good compromise between the revision quality (minimality) and efficiency. Although we are focused on the flooding problem, we hope extend this work to revise any problem expressed as a linear constraint network.

Acknowledgments. This work has been supported by the REV!GIS project, IST-1999-14189. We thank our colleagues in the project for the many discussions that we had on the flooding problem. Especially, N. Wilson, R. Jeansoulin, Y. Lassoued, and E. Würbel.

REFERENCES

[1] C. Alchourròn, P. Gärdenfors, and D. Makinson, 'On the logic of theory change : Partial meet functions for contraction and revision', *Journal of Symbolic Logic*, **50**, 510–530, (1985).

[2] A. Darwiche and J. Pearl, 'On the logic of iterated revision', *Artificial Intelligence*, **89**, 1–29, (1997).

[3] H. Katsuno and A.O. Mendelzon, 'Propositional knowledge base revision and minimal change', *Artificial Intelligence*, **52**, 263–294, (1991).

[4] *Constraint Databases*, eds., G. Kuper, G. L. Libkin, and J. Paradaens, Springler-Verlag, Berlin, Heidelberg, New York, 2000.

[5] C.E. Leiserson and J.B. Saxe, 'A mixed-integer linear programming problem which is efficiently solvable', in *Proc. of the 21st Allerton conf. on Communications, Control, and Computing*, pp. 204–213, (1983).

[6] Y.Z. Lia and C.K. Wong, 'An algorithm to compact a vlsi symbolic layout with mixed constraints', in *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, volume 2, pp. 62–69, (1983).

[7] D. Raclot and C. Puech, 'Photographies aériennes et inondations : Globalisation d'informations floues par un système de contraintes pour définir les niveaux d'eau en zone inondée', *Revue Internationale de Géomatique*, **8**, 191–206, (1998).

[8] D. Raclot and C. Puech, 'What does ai bring to hydrology? the example of aerial pictures used to determine flooding levels', in *ECSQARU-2001 workshop on Spatio-Temporal Reasoning and Geographic Information Systems*, (2001).

[9] P. Rigaux, M. Scholl, and A. Voisard, *Spatial databases with application to GIS*, Morgan Kaufmann, 2002.

[10] R. Shostak, 'Deciding linear inequalities by computing loop residues,', *Journal of ACM*, **28**(4), 769–779, (1981).

[11] E. Würbel, R. Jeansoulin, and O. Papini, 'Revision : an application in the framework of gis', in *Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning (KR'00)*, (2000).

[12] E. Würbel, R. Jeansoulin, and O. Papini, 'Spatial information revision: a comparison between 3 approaches', in *6th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU4'01*, eds., S. Benferhat and P. Besnard, pp. 454–465, Toulouse, France, (September, 19-21 2001).