Révision d'informations géographiques à base de contraintes temporelles

Mahat Khelfallah
LSIS
CMI - Université de Provence
13453 Marseille
mahat@cmi.univ-mrs.fr

Bélaid Benhamou
LSIS
CMI - Université de Provence
13453 Marseille
benhamou@cmi.univ-mrs.fr

Robert Jeansoulin
LSIS / CNRS
CMI - Université de Provence
13453 Marseille
jeansoul@cmi.univ-mrs.fr

Résumé *La multiplicité des sources de l'information, son caractère incomplet et/ou incertain provoquent l'apparition de plusieurs incohérences. La révision est l'opération qui consiste à identifier ces incohérences et à les « corriger » en rétractant le moins d'informations possible. Dans ce papier, nous nous intéressons à la révision d'informations géographiques dans le cadre du problème de l'inondation. Nous montrons comment représenter et réviser ce problème en utilisant le formalisme temporel STP (Simple Temporal Problems) basé sur les contraintes linéaires. Nous proposons trois stratégies de révision basées sur la résolution de contraintes linéaires. Enfin, nous appliquons ces trois approches au problème réel d'inondation de la vallée de l'Hérault et nous comparons nos résultats par rapport à des travaux antérieurs.*
mot-clés *: Révision, Simple Temporal Problems STP, Contraintes temporelles, Informations géographiques, Contraintes linéaires.*

1 Introduction

Les informations géographiques proviennent généralement de plusieurs sources et sont très souvent incomplètes et / ou incertaines et l'ajout de nouvelles informations provoque très souvent l'apparition d'incohérences. Si les nouvelles informations sont plus fiables que les anciennes, ces dernières sont révisées pour rétablir la cohérence de l'ensemble des informations. La révision identifie d'abord les incohérences, puis les corrige tout en conservant le maximum d'informations initiales, autrement dit, en rétractant le minimum d'informations possible.

De nombreuses recherches ont été menées dans le domaine de la révision des croyances [AGM 85, KM 91, DP 97]. L'un des travaux de base dans ce domaine est celui d'Alchourròn, Gärdenfors et Makinson [AGM 85] où le fameux paradigme AGM a été établi comme un ensemble de propriétés que doit vérifier un opérateur de révision.

Les stratégies de révision obéissent généralement à deux principes :

– *La primauté des nouvelles informations sur les anciennes*, i.e., les nouvelles informations sont plus fiables que les anciennes, et ce sont celles-ci qui doivent être révisées ;
– *La politique du changement minimal*, et qui garantit la « rétraction » d'un nombre minimum d'informations.

L'application des méthodes de révision à des problèmes réels est souvent très difficile, voire impossible, vu la grande complexité des algorithmes de révision. Cet inconvénient devient plus important encore lorsque ces applications manipulent un grand volume de données, comme c'est le cas dans les applications géographiques. Cependant, les informations géographiques s'expriment souvent par des contraintes spatiales et / ou temporelles qui peuvent parfois être modélisées par des contraintes linéaires [RSV 02, KLP 00].

Dans ce papier, nous nous intéressons à la révision de connaissances basée sur la résolution de contraintes linéaires dans le cadre du formalisme temporel STP (Simple Temporal Problem). Nous proposons trois stratégies différentes de révision :

– *La révision conflit par conflit* dans laquelle la correction s'effectue à chaque détection d'une incohérence, jusqu'à la restauration de la cohérence totale du problème.
– *La révision de toutes les incohérences à la fois*, et qui consiste à identifier dès le départ toutes les incohérences du problème, puis à calculer le plus petit sous-ensemble de contraintes dont la révision suffit au rétablissement de la cohérence.
– *La révision hybride*, qui consiste à identifier une partie représentative des différents conflits, les corriger (en identifiant la meilleure correction par rapport au nombre de contraintes modifiées), puis à répéter ce processus jusqu'à élimination de tous les conflits.

Nous appliquons et testons ces approches sur les données d'un problème réel dans le domaine de l'hydraulique et qui est le problème de crue dans la vallée de l'Hérault, et nous comparons nos résultats avec ceux obtenus dans des travaux précédents [WJP 00, WJP 01, WKJ 03, RP 98, RP 01].

Ce papier est organisé comme suit. Dans la section 2, nous introduisons le formalisme STP qui représente le cadre théorique de notre étude. Dans la section 3, nous décrivons le problème d'inondation ainsi que sa représentation comme un STP. Dans la section 4, nous présentons un algorithme de détection de conflits dans un STP. Dans la section 5, nous définissons trois stratégies de révision : la révision conflit par conflit, la révision globale et la révision hybride. Dans la section 6, nous testons ces approches sur les données du problème de l'inondation de l'Hérault et nous présentons les résultats obtenus pour chacune d'elles. Nous comparons, dans la section 7, nos résultats avec des résultats obtenus dans des travaux précédents sur le problème de l'inondation, avant de conclure dans la section 8.

2 Préliminaires

Avant de présenter notre méthode de révision à base de STP, nous devons rappeler brièvement ce qu'est le formalisme STP ainsi que certaines de ses propriétés. Pour plus de détails, voir Dechter et al. [DPM 91].

Un STP (pour Simple Temporal Problem) est défini par la structure (V, C) où V est un ensemble fini de variables $X_1, ..., X_n$, à domaines réels continus représentant à l'origine des points du temps, et C est l'ensemble des contraintes unaires ou binaires liant ces variables. Une contrainte unaire $C_i = [a_i, b_i]$ définie sur la variable X_i restreint son domaine à l'intervalle $[a_i, b_i]$. C_i est exprimée par les deux inéquations : $a_i \leq X_i \leq b_i$. La contrainte binaire $C_{ij} = [a_{ij}, b_{ij}]$ restreint la valeur de la distance $X_j - X_i$ à l'intervalle $[a_{ij}, b_{ij}]$. La contrainte C_{ij} est exprimée par les deux inégalités suivantes : $a_{ij} \leq X_j - X_i \leq b_{ij}$.

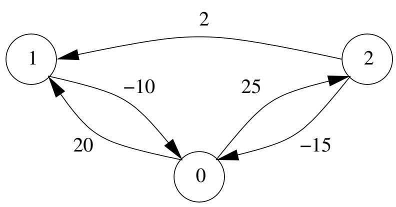
Une contrainte unaire peut être transformée en une contrainte binaire par l'introduction d'une nouvelle variable X_0 représentant l'origine du temps. On choisit souvent zéro comme origine du temps ($X_0 = 0$). Dans ce cas la contrainte unaire $C_i : a_i \leq X_i \leq b_i$ définie sur X_i est équivalente à la contrainte binaire $C_{i0} : a_i \leq X_i - X_0 \leq b_i$ définie sur les variables X_i et X_0.

Un tuple $X = (x_1, ..., x_n)$ de valeurs réelles est dit solution du STP si l'instanciation $X_1 = x_1, ..., X_n = x_n$ satisfait toutes les contraintes de ce dernier. Un STP est cohérent s'il possède au moins une solution.

Exemple Soit le STP $P = (V, C)$ où l'ensemble des variables $V = \{X_1, X_2\}$ et l'ensemble des contraintes $C = \{10 \leq X_1 \leq 20, 15 \leq X_2 \leq 25, X_1 - X_2 \leq 2\}$. Le tuple $x = (20, 25)$ est une solution du STP P. Par conséquent, ce dernier est cohérent.

On associe à un STP un graphe orienté et pondéré, $G_d = (V, E_d)$, appelé graphe des distances, où V est l'ensemble des sommets représentant les variables du STP, et E_d est l'ensemble des arcs $i \to j$, étiquetés par a_{ij}, représentant l'inéquation $X_j - X_i \leq a_{ij}$. Le graphe des distances associé au STP précédent est donné dans la figure 1.

FIG. 1: Graphe des distances



Nous sommes maintenant en mesure de décrire et de formaliser le problème de l'inondation qui servira comme exemple d'illustration pour nos techniques de révision dans le formalisme des STP.

3 Représentation du problème de l'inondation par un STP

Nous décrivons dans cette section le problème de l'inondation de la vallée de l'Hérault que nous formalisons sous forme de problème STP.

3.1 Description du problème

Lors d'une inondation de l'Hérault, une partie de la vallée a été étudiée pour trouver une estimation correcte de la hauteur d'eau à partir de photographies aériennes et de connaissances topographiques de la région.

La partie étudiée a été divisée en n parcelles, assez petites pour considérer que les niveaux d'eau sont constants dans chacune d'elle.

Deux sources d'information sont disponibles. La première source concerne des estimations sur les niveaux d'eau dans chaque parcelle. Chaque estimation du niveau d'eau dans une parcelle est donnée par un intervalle des valeurs possibles et est représentée par une valeur minimale et une valeur maximale.

La seconde source représente des observations sur les relations hydrauliques existant entre certaines parcelles adjacentes. Ces relations peuvent être soit des relations de flux, ou bien des relations d'équilibre entre deux parcelles adjacentes. Un flux allant d'une parcelle i vers une parcelle j est observé si le niveau d'eau dans i est au moins supérieur au niveau d'eau dans la parcelle j. Un équilibre hydraulique entre deux parcelles se traduit par une égalité entre leur niveau d'eau. Pour une description plus détaillée, voir [RP 98, RP 01].

Les deux sources d'information étant contradictoires et la source relative aux observations sur les relations hydrauliques étant plus fiable que celle relative aux estimations des hauteurs d'eau dans les parcelles, les estimations des niveaux d'eau sont *révisées* par les relations hydrauliques.

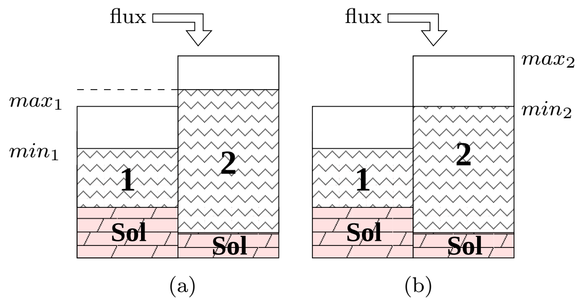
Dans la figure 2, nous illustrons un petit exemple de contradiction pouvant apparaître en considérant nos deux sources d'informations. Le schéma (a) représente une contradiction entre les niveaux d'eau dans les parcelles 1 et 2 avec le flux observé allant de 1 vers 2. Dans le schéma (b) la valeur minimale du niveau d'eau dans la parcelle 2 a été révisée.

Nous allons voir dans la suite de cet article comment détecter ce genre de contradiction ou conflit et comment les corriger. Mais avant, il faut représenter les informations de notre problème dans le formalisme des STP.

3.2 Représentation du problème

La représentation de ce problème par des contraintes linéaires est immédiate. Nous associons à chaque parcelle i une variable X_i qui représente le niveau d'eau

FIG. 2: Exemple de contradiction



dans cette parcelle. Les deux sources d'information sont représentées comme suit :

Les estimations sur les niveaux d'eau : A chaque parcelle i, est associée la contrainte C_i : $min_i \leq X_i \leq max_i$, où min_i et max_i sont les bornes de l'intervalle d'estimation du niveau d'eau dans la parcelle i.

Les observations sur les relations hydrauliques : Chaque flux observé allant de la parcelle i vers la parcelle j est traduit par la contrainte $X_j - X_i \leq 0$; et tout équilibre observé entre les parcelles i et j se traduit par les contraintes : $X_j - X_i \leq 0$ et $X_i - X_j \leq 0$.

Le problème d'inondation est donc un STP $P = (V, C)$ où V est l'ensemble des variables associées aux niveaux d'eau des différentes parcelles et C l'ensemble des contraintes représentant d'une part les estimations des niveaux d'eau dans ces parcelles et d'autre part les relations de flux existant entre elles.

Après avoir modélisé le problème, nous nous intéressons à la recherche des incohérences à base desquelles nous révisons l'information, c'est l'objet de la section suivante.

4 Détection des incohérences

Nous présentons une méthode qui détecte les incohérences dans l'ensemble des contraintes. Pour cela, nous utilisons un résultat important dans les STP qui stipule qu'un *STP est cohérent si et seulement si son graphe des distances ne contient pas de circuits négatifs* [1] [Sho 81, LW 83, LS 83].

Ainsi, la présence d'un circuit négatif dans le graphe des distances est synonyme de conflit dans l'ensemble des contraintes du STP, et par conséqunet, chercher les incohérences dans un STP revient à rechercher les circuits négatifs dans son graphe des distances.

Une des méthodes efficaces pour tester si un graphe contient des circuits négatifs est d'appliquer au graphe des distances l'algorithme de Floyd-Warshall

[1] Un circuit négatif est un circuit dont la somme des étiquettes de ses arcs est inférieure à 0.

[GM 79] qui calcule les distances des plus courts chemins entre des paires de sommets du graphe. Cet algorithme donne des témoins d'existence de circuits négatifs si le problème est incohérent, mais n'identifie pas les circuits négatifs qui représentent les incohérences existant dans le problème. Comme la révision du problème est basée sur le traitement des incohérences, nous étendons l'algorithme de Floyd-Warshall pour calculer les circuits négatifs dans le graphe des distances. Pour se faire, nous associons à chaque couple de sommets (i, j) du graphe, une variable $chemin_{ij}$ qui représente le plus court chemin allant de i vers j, et une variable d_{ij} qui représente la distance du chemin $chemin_{ij}$. Cette variable est initialisée à l'arc $i \rightarrow j$ s'il existe, ou à vide sinon. Ci-dessous la procédure codant la méthode.

Calcul des plus courts chemins et leur distance (Algorithme de Floyd-Warshall étendu FWe)

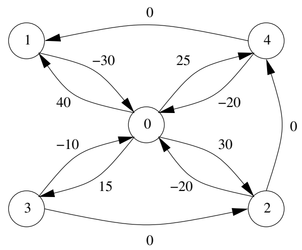
Début
{Initialisation des variables}
    Pour $i := 1$ à n faire
        $d_{ii} := 0$
        $chemin_{ii} := \emptyset$
    Pour $i, j := 1$ à $n (i \neq j)$ faire
        si ($X_j - X_i \leq a_{ij}$ est une contrainte du problème) alors
            $d_{ij} := a_{ij}$
            $chemin_{ij} := (i, j)$
        sinon $d_{ij} := \infty$
            $chemin_{ij} := \emptyset$
{Calcul des plus courts chemins et leur distance}
    pour $k := 1$ à n faire
        pour $i, j := 1$ à n faire
            si $((d_{ij} > d_{ik} + d_{kj})$ et $(chemin_{ik} + chemin_{kj})$ est un chemin/circuit élémentaire[2]) alors
                $d_{ij} := d_{ik} + d_{kj}$
                $chemin_{ij} := chemin_{ik} + chemin_{kj}$
                si($i = j$ et $d_{ii} < 0$) alors Sauvegarder le *circuit négatif*
Fin

Les circuits négatifs sont identifiés par la présence d'une distance négative sur un élément de diagonale d_{ii}. Néanmoins, cet algorithme ne donne pas « tous » les circuits négatifs pouvant exister dans le graphe car nous ne considérons que les plus courts chemins. L'exemple suivant illustre ce fait.

Exemple. Nous considérons un ensemble de quatre parcelles dont les niveaux d'eau sont représentés par l'ensemble de variables $\{X_1, X_2, X_3, X_4\}$. Les contraintes sur les estimations des niveaux d'eau et flux observés entre ces parcelles induisent le graphe des distances de la figure 3.

[2]Un chemin élémentaire est un chemin qui passe au plus une fois par un même sommet. Un circuit élémentaire est un circuit qui passe exactement une fois par chacun de ses sommets.

Fig. 3: Graphe des distances



L'application de l'algorithme de calcul des plus courts chemins détecte les circuits négatifs suivants : [0 , 3 , 2 , 0] , [0 , 3 , 2 , 4 , 1 , 0], [1 , 0 , 3 , 2 , 4 , 1] , [2 , 0 , 3 , 2] , [2 , 4 , 1 , 0 , 3 , 2] , [3 , 2 , 4 , 1 , 0 , 3] , [3 , 2 , 0 , 3] , [4 , 1 , 0 , 4] , [4 , 1 , 0 , 3 , 2 , 4].

Nous pouvons remarquer que le circuit négatif [3 , 2 , 4 , 0 , 3] n'a pas été détecté. Cet exemple illustre bien l'incomplétude de l'algorithme précédent pour le calcul de tous les circuits négatifs.

Dans la section suivante, nous allons montrer comment il est utilisé dans trois différentes approches de révision.

5 Approches de révision

Avant de présenter les algorithmes des différentes approches que nous avons considérées, nous définissons le principe général de notre révision.

5.1 Principe général

La révision des contraintes consiste à :

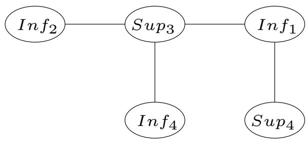
1. Identifier les conflits, c'est à dire les circuits négatifs dans le graphe des distances,
2. Calculer, à partir des circuits négatifs détectés, un plus petit sous-ensemble de contraintes « conflictuelles » dont la correction suffit à rétablir la cohérence,
3. Corriger ces contraintes.

La première étape est assurée par l'algorithme de Floyd-Warshall étendu décrit précédemment, il reste à définir les deux autres étapes.

5.1.1 Détection et représentation des conflits

Nous rappelons que nous nous intéressons à la révision des estimations des niveaux d'eau dans les parcelles par les relations de flux existant entre ces parcelles. Les estimations des niveaux sont représentées par des contraintes

FIG. 4: Graphe des conflits



unaires de la forme $min_i \leq X_i \leq max_i$, qui se transforment en $min_i \leq X_i - X_0 \leq max_i$ après l'introduction de la variable X_0 représentant l'origine ($X_0 = 0$).

On comprend alors que les contraintes conflictuelles sont celles qui sont représentées par des arcs dont l'une des extrémités est le sommet 0. Un arc $0 \rightarrow i$ (respectivement un arc $i \rightarrow 0$) représente la contrainte $X_i - X_0 \leq max_i$ (respectivement la contrainte $X_0 - X_i \leq -min_i$).

Comme les circuits calculés sont élémentaires, il existe exactement une contrainte sur la valeur maximale d'un niveau d'eau X_i et une autre sur la valeur minimale d'un niveau d'eau X_j dans un conflit.

Si l'on considère l'ensemble de sommets $S = \{Inf_1, Sup_1, \ldots, Inf_i, Sup_i, \ldots, Inf_n, Sup_n\}$ correspondant aux bornes inférieures et supérieures (un sommet par borne) des niveaux d'eau dans les parcelles conflictuelles, alors tout conflit entre la borne inférieure de la variable X_i et la borne supérieure de la variable X_j est exprimé par l'arête (Inf_i, Sup_j). Soit E l'ensemble de ces arêtes, on représente alors les conflits détectés par le graphe $G_c = (S, E)$ appelé *graphe des conflits*.

Exemple. Nous reprenons l'exemple précédent. Nous identifions dans le circuit négatif [0 , 3 , 2 , 4 , 1 , 0], les contraintes conflictuelles $X_3 - X_0 \leq 15$ et $X_0 - X_1 \leq -30$. Autrement dit, la borne supérieure du niveau d'eau dans la parcelle 3 est en contradiction avec la borne inférieure du niveau d'eau dans la parcelle 1. Ceci rajoute l'arête (Sup_3, Inf_1) dans le graphe des conflits. En considérant tous les circuits négatifs, nous obtenons le graphe des conflits de la figure 4.

5.1.2 Calcul d'un plus petit sous-ensemble de contraintes à réviser

Pour supprimer les conflits, il faut corriger certaines contraintes impliquées dans ces derniers. Mais le choix de ces contraintes n'est pas évident. Il dépend, en réalité, de la politique de révision adoptée.

Dans ce travail, nous nous intéressons à la politique de changement minimal. Plus précisément, nous voulons relaxer un nombre minimum de contraintes pour supprimer tous les conflits détectés.

Chercher un sous-ensemble minimal (par rapport à la cardinalité) de contraintes à corriger revient alors à chercher un sous-ensemble de sommets du graphe des conflits dont la suppression provoque la suppression de toutes les arêtes de ce graphe. Cette opération correspond à la recherche d'un *transversal de cardinalité minimale du graphe des conflits*. Nous rappelons la définition d'un transversal.

Définition 1. Soit G un graphe défini par $G = (V, E)$. T est un *transversal* de G si et seulement si T est un sous-ensemble de l'ensemble des sommets V, et pour toute arête (x, y) de E, $T \cap \{x, y\} \neq \emptyset$.

Définition 2. Un transversal T d'un graphe G est un transversal de cardinalité minimale si et seulement si pour tout transversal T' de G, si $cardinalit\acute{e}(T') \leq cardinalit\acute{e}(T)$ alors $T = T'$.

Exemple. Le graphe des conflits de l'exemple précédent possède deux transversaux de cardinalité minimale : $\{Sup_3, Inf_1\}$ et $\{Sup_3, Sup_4\}$. Par conséquent, la rétraction des contraintes relatives à Sup_3 et Inf_1 (respectivement, relatives à Sup_3 et Sup_4) est suffisante pour éliminer tous les conflits.

Il existe beaucoup de travaux sur la recherche de transversaux dans un graphe [EG 91]. Ce problème est dans la classe des problèmes NP-complets. Notre méthode de recherche de transversal de cardinalité minimale est basée sur un algorithme de backtracking intelligent défini sur les sommets du graphe des conflits. Ce dernier ne calcule pas systématiquement tous les transversaux du graphe des conflits, mais abandonne toute branche de l'arbre de recherche dès que la cardinalité du transversal partiel associé dépasse celle du transversal de référence jusque-là calculé.

Il est clair que la complexité théorique de cet algorithm par rapport au nombre de sommets du graphe est exponentielle dans le pire des cas. Mais, dans la pratique nous avons obtenu de très bons résultats grâce à la coupure de branches inutiles dans l'arbre de recherche et au fait que le nombre de sommets du graphe des conflits (i.e., le nombre de contraintes conflictuelles) est très petit par rapport au nombre total de contraintes du problème initial.

5.1.3 Correction des contraintes

Nous avons jusqu'à présent, décrit comment détecter les conflits du problème, et comment identifier un sous-ensemble de contraintes dont la correction suffit pour éliminer ces conflits et pour rétablir la cohérence du problème. Nous nous intéressons maintenant à la manière dont sera effectuée cette correction.

Soit un conflit entre les deux contraintes $X_i \leq b_i$ et $a_j \leq X_j$. L'élimination de ce conflit nécessite la correction de soit la borne supérieure de X_i, ou bien de la borne inférieure de X_j. [3]

Soit d_{ii} la distance calculée du circuit négatif qui a permis l'identification de ce conflit. Remplacer la contrainte $X_i \leq b_i$ (respectivement, la contrainte $a_j \leq X_j$) par la contrainte $X_i \leq b_i - d_{ii}$ (respectivement, par la contrainte $a_j + d_{ii} \leq X_j$) éliminera le conflit initial.

[3] La correction des deux bornes n'est pas traitée par souci de minimisation du nombre de valeurs révisées.

Exemple. Le conflit détecté dans l'exemple précédent entre $X_3 - X_0 \leq 15$ et $X_0 - X_1 \leq -30$ (équivalente à $X_1 \geq 30$), peut être corrigé soit en augmentant la borne supérieure de X_3 de 15 ou bien en baissant la borne inférieure de X_1 de 15 également. Si l'on choisit la première solution, on obtient alors la contrainte corrigée $X_3 - X_0 \leq 30$.

La correction d'une contrainte ne peut provoquer de nouveaux conflits car celle-ci est en réalité un élargissement d'un intervalle donc un affaiblissement de contraintes.

Nous présentons maintenant les trois méthodes de révision que nous avons mises en œuvre. Elles sont basées sur trois stratégies différentes :

1. Réviser un conflit dès sa détection, et relancer cette opération jusqu'à la cohérence,
2. Détecter tous les conflits puis les corriger, ou
3. Une stratégie intermédiaire qui détecte un ensemble de conflits, les révise puis relance le processus jusqu'à ce qu'il n'y ait plus d'incohérences.

Nous commençons par décrire la méthode de révision conflit par conflit.

5.2 Révision conflit par conflit

L'idée la plus intuitive est de corriger les conflits dès leur détection. Autrement dit, dès que des contraintes conflictuelles sont identifiées, elles sont révisées. Ce processus est répété jusqu'à l'élimination de tous les conflits.

Algorithme de révision conflit par conflit
```
Début :
   Rechercher un circuit négatif en appliquant l'algorithme Fwe [4].
   Si un circuit négatif est détecté alors
      Identifier les contraintes conflictuelles du circuit détecté
      Choisir une des contraintes identifiées et la corriger
      Aller à Début
   Sinon Aller à Fin
Fin
```

Cet algorithme termine car la révision des conflits détectés ne peut en générer de nouveaux et que le nombre de conflits est fini. De plus, cet algorithme effectue au plus $(k+1)$ itérations pour rétablir la cohérence, où k est le nombre total des conflits dans l'ensemble de contraintes initial.

Cet algorithme ne garantit pas une révision minimale puisqu'aucune optimisation du nombre de contraintes corrigées n'a été effectuée. Mais son intérêt n'est pas moindre surtout dans les applications où la détection de tous les conflits est très difficile voire impossible.

[4] Dès la détection d'un circuit négatif, l'algorithme FWe est arrêté.

5.3 Révision de tous les conflits

Contrairement à la méthode précédente, cette approche respecte le critère de minimalité de changement pour réviser l'information. Elle calcule d'abord tous les conflits du problèmes relatifs aux circuits négatifs détectés, ensuite les corrige en proposant une solution minimale en terme de nombre de contraintes à réviser.

Le problème est que l'algorithme de Floyd-Warshall étendu présenté dans la section précédente ne détecte pas " tous " les circuits négatifs et par conséquent ne détecte qu'une partie des conflits existant entre les contraintes. Il est alors nécessaire de remédier à ce problème pour mettre en oeuvre cette méthode. Pour se faire, nous utilisons le résultat de la proposition suivante qui fournit un moyen de calcul efficace de tous les conflits

Proposition. Soit G_d le graphe des distances du STP. Il existe un circuit négatif passant par le sommet i si et seulement si la distance d_{ii} calculée par l'algorithme FWe est négative ($d_{ii} < 0$).

Preuve Puisque l'algorithme FWe calcule les plus courts chemins entre deux sommets quelconques, il calculera, en particulier, le plus court chemin allant du sommet i vers i. S'il existe un circuit négatif dans le graphe des distances G_d passant par le sommet i de longueur $l_i < 0$ alors l'algorithme FWe calculera la distance du plus court chemin. Comme la distance d_{ii} est minimale alors, $d_{ii} < l_i < 0$. La preuve dans l'autre sens est évidente.

Nous pouvons ainsi trouver les circuits négatifs "manquants" en effectuant une recherche exhaustive des circuits négatifs impliquant au moins un sommet i tel que $d_{ii} < 0$. Nous pouvons à présent décrire l'algorithme de la révision de tous les conflits.

Algorithme de révision de tous les conflits

Début

Appliquer l'algorithme FWe au graphe des distances

Rechercher tous les circuits négatifs non détectés dans l'étape précédente

Identifier les contraintes conflictuelles des circuits détectés

Calculer un plus petit sous-ensemble de contraintes à réviser (transversal de cardinalité minimale)

Corriger les contraintes de ce sous-ensemble

Fin

Le principe de changement minimal par rapport au nombre de bornes corrigées est garanti par le calcul d'un plus petit sous-ensemble de contraintes à réviser. L'avantage de cette méthode est qu'elle fournit une révision globale et minimale. Son inconvénient est que la recherche exhaustive de tous les circuits négatifs peut s'avérer très coûteuse en temps de calcul et en espace. Elle est exponentielle en nombre de sommets dans le pire des cas. Mais dans la pratique on peut toujours espérer que le nombre de circuits négatifs soit relativement petit et que nous puissions le calculer efficacement.

La dernière méthode que nous présentons est une hybridation des deux méthodes précédentes. Elle est considérée comme une méthode intermédiaire entre l'approche de révision conflit par conflit et l'approche de révision de tous les conflits.

5.4 Révision hybride

L'approche de la révision hybride consiste à rechercher un ensemble de conflits (qui n'est pas forcément total et exhaustif), puis les réviser. Ces deux étapes sont répétées jusqu'à restauration totale de la cohérence.

La recherche des conflits est assurée par l'algorithme *FWe*. En effet, on garde en mémoire un circuit négatif pour chaque diagonale indiquant une incohérence. La révision des contraintes conflictuelles identifiées est relativement minimale par rapport au nombre de bornes corrigées.

Algorithme de révision hybride

Début :
    Appliquer l'algorithme FWe au graphe des distances
    Si des circuits négatifs sont détectés alors
        Identifier les contraintes conflictuelles associées
        Calculer un plus petit sous-ensemble de contraintes à réviser
        Corriger les contraintes de ce sous-ensemble
        Aller à Début
    Sinon Aller à Fin
Fin :

Cet algorithme termine car la révision des conflits détectés ne peut en générer d'autres et que le nombre de conflits est fini. De plus, l'algorithme procède au plus en $(d + 1)$ itérations (où d est le nombre maximal de conflits impliquant la même contrainte).

Il n'est pas rare que dès la première itération, tous les conflits soient éliminés. Cela peut se produire lorsque la correction d'une borne se propage et élimine d'autres conflits non encore détectés. D'ailleurs, c'est l'un des points forts de cette méthode. Elle effectue moins d'itération que l'approche de la révision conflit par conflit tout en révisant mieux. Cependant, elle révise moins bien que la méthode de la révision de tous les conflits, mais elle serait plus rapide car cette approche n'a pas le souci de calculer tous les circuits négatifs qui est le point faible de la méthode qui génère tous les conflits.

6 Résultats expérimentaux

Les trois algorithmes de révision présentés dans ce papier ont été mis en oeuvre et testés sur des données réelles d'une application d'inondation dans la vallée de l'Hérault.

Tab. 1: Résultats expérimentaux

Algorithmes de révision	conflit par conflit	tous les conflits	hybride
Nombre d'itérations	12	1	2
Nombre de conflits détectés	11	15	14
Nombre de contraintes corrigées	11	10	10
Temps d'exécution (s)	14	3	3

L'application contient environs 300 parcelles, mais seules 180 possèdent des estimations sur leur hauteur d'eau. A chacune de ces 180 parcelles, sont associées une variable qui représente son niveau d'eau et deux contraintes qui représentent l'intervalle des valeurs possibles de ce niveau. Donc, nous avons 360 contraintes d'estimation de bornes d'intervalles. A cela s'ajoutent, environs 270 contraintes représentant les relations de flux et d'équilibre entre les parcelles. Ainsi, notre problème est à 180 variables et environ 630 contraintes.

L'application des trois algorithmes de révision a donné les résultats présentés dans le tableau 1.

Nous remarquons que l'algorithme de révision conflit par conflit n'a pas eu besoin de faire les 15 itérations pour corriger tous les conflits existants. 12 itérations ont suffit. Ceci est du à la propagation des corrections qui ont pu éliminer des conflits jusque là non détectés.

La même remarque peut être faite concernant la révision hybride. Bien que la première itération n'ait pas détecté tous les conflits existants (elle en a omis un), la correction a pu rétablir la cohérence générale en une seule itération.

Nous remarquons que le nombre de contraintes corrigées est inférieur au nombre de conflits. Il est le même dans les deux approches : la révision de tous les conflits et la révision hybride. La raison est que dans la révision hybride, nous minimisons le nombre de contraintes corrigées et que presque tous les conflits du problème ont été traités.

Concernant le temps d'exécution, c'est la révision conflit par conflit qui prend le plus de temps (14 seconces) vu le nombre élevé d'itérations nécessaires à la restauration de la cohérence. Les deux autres algorithmes prennent chacun 3 secondes.

7 Comparaison avec des travaux antérieurs

Würbel et al. [WJP 00, WJP 01] ont proposé plusieurs méthodes pour la révision des informations géographiques dans le contexte du problème d'inondation. Les meilleurs résultats ont été obtenus par la méthode REMr, basée sur l'algorithme de Reiter pour le diagnostic [Rei 87] et qui n'a pu traité qu'une partie constituée de 12 parcelles.

Une solution a été donnée par Wilson et al. [Wil 02, WKJ 03] basée sur la propagation des contraintes linéaires. Cette méthode est très proche de la stratégie de révision globale (de tous les conflits) que nous avons défini dans ce papier.

Les travaux de Raclot et Puech [RP 98, RP 01] sur l'application ont permis la résolution du problème. La méthode qu'ils ont proposée est proche de la révision conflit par conflit que nous proposons dans ce papier. La différence est qu'ils resserrent, dans leur méthode, les intervalles des estimations pour optimiser les solutions, ce qui s'éloigne un peu de la révision.

8 Discussion et Conclusion

Dans cet article, nous avons présenté trois approches de révision pour les informations géographiques exprimées sous forme de réseaux de contraintes STP.

Bien que nous nous sommes focalisés sur le problème d'inondation, ces méthodes peuvent être appliquées pour réviser tout problème représentable sous forme d'un STP. Nous pensons notamment à certains problèmes d'ordonnancement de tâches et de scheduling en raisonnement temporel.

Les trois méthodes adoptent des stratégies différentes. La première, procède conflit par conflit, elle s'arrête au premier conflit détecté, le corrige et relance le processus. Cette approche a l'avantage d'être simple à mettre en oeuvre et peut s'avérer efficace pour les problèmes présentant un grand nombre de conflits interdépendants . L'inconvénient de cette approche est sa qualité de révision, elle ne respecte pas le critère de minimalité et peut nécessiter un grand nombre d'itérations pour réviser un problème. L'étude expérimentale menée sur le problème de l'inondation montre ce fait. Elle est la méthode la moins efficace bien que certains conflits aient été indirectement corrigés par la révision d'autres conflits.

La deuxième approche proposée est celle qui calcule tous les conflits, et qui corrige un nombre minimal de contraintes conflictuelles pour restaurer la cohérence. Cette méthode obéit au principe du changement minimal par rapport au nombre de corrections portées sur les contraintes. Elle a l'avantage de garantir une bonne révision et sera efficace pour des problèmes présentant un nombre raisonnable de conflits comme c'est le cas pour le problème de l'inondation.

La troisième méthode est une hybridation des deux premières. Elle détecte un sous-ensemble de conflits, ensuite les corrige, et répète le processus jusqu'à la restauration de la cohérence. Elle réalise ainsi un bon compromis entre la qualité de révision et l'efficacité en temps de calcul. Cette approche a donné les mêmes résultats que l'approche globale car le sous-ensemble de conflits qu'elle considère à la première itération est presque total. Mais il est certain que pour d'autres problèmes, les résultats peuvent être différents.

Le problème de l'inondation est assez simple et ne peut donc fournir d'arguments complets pour comparer les trois méthodes. Nous pensons les expérimenter prochainement sur des instances de problèmes générés aléatoirement et sur d'autres problèmes réels pour connaître leurs performances et leurs limites.

Remerciements

Ce travail a été financé par le projet REV!GIS, IST-1999-14189. Nous remercions nos collègues du projet pour les nombreuses discussions que nous avons eu sur le problème de l'inondation. En particulier, Nic Wilson, Yassine Lassoued, Eric Würbel, Christian Puech et Damien Raclot.

Références

[AGM 85] Alchourròn, C., Gärdenfors, P., Makinson, D. : *On the logic of theory change : Partial meet contraction and revision functions.* Journal of Symbolic Logic. 50(2) (1985) 510–530.

[DP 97] Darwiche, A., Pearl, J. : *On the logic of iterated belief revision.* Artificial Intelligence. **89** (1997) 1–29.

[DPM 91] Dechter, R., Meiri, I., and Pearl, J., *Temporal constraint networks.* Artificial Intelligence. **49** (1991) 61–95.

[EG 91] T. Eiter and G. Gottlob : *Hypergraph Transversal Computation and Related Problems in Logic and AI.* JELIA 2002 : (2002) pp549-564.

[GM 79] Gondron, M., Minoux, M. : *Graphes et algorithmes.* Ed Eyrolles, 1979.

[KM 91] Katsuno, H., Mendelzon, A. O. : *Propositional knowledge base revision and minimal change.* Artificial Intelligence. **52** (1991) 263–294.

[KLP 00] Kuper, G., Libkin, G. L., Paradaens, J. (eds.) : *Constraint Databases.* Springler-Verlag, Berlin, Heidelberg, New York, (2000).

[LS 83] Leiserson C.E., Saxe, J.B. : *A mixed-integer linear programming problem which is efficiently solvable.* in Proc. of the 21st annual Allerton conference on Communications, Control, and Computing (1983) 204-213.

[LW 83] Lia, Y.Z., Wong, C.K. : *An algorithm to compact a VLSI symbolic layout with mixed constraints.* IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems 2 (2) (1983) 62-69.

[RP 98] Raclot, D., Puech, C. : *Photographies aériennes 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)** (1998) 191–206.

[RP 01] Raclot, D., and Puech, C. : *What does AI Bring to Hydrology ? The Example of Aerial Pictures used to Determine Flooding Levels.* Report associated with invited lecture by Christian Puech at ECSQARU-2001 workshop on Spatio-Temporal Reasoning and Geographic Information Systems (2001).

[Rei 87] Reiter, R. : *A theory of diagnosis from first principles.* Artificial Intelligence, **32** (1987) 57–95.

[RSV 02] Rigaux, P., Scholl, M., Voisard, A. : *Spatial databases with application to GIS.* Morgan Kaufmann Publishers (2002).

[Sho 81] Shostak, R. : *Deciding linear inequalities by computing loop residues.* J. ACM 28 (4) - (1981), 769-779.

[Wil 02] Wilson, N. : *The logic of linear constraints and its application to the flooding Problem.* REVIGIS project report (2002).

[WKJ 03] Wilson, N., Khelfallah, M., Jeansoulin, R. : *Geographic information revision based on linear constraints.* REV!GIS project report, 2003.

[WJP 00] Wurbel, E., Jeansoulin, R., Papini, O. : *Revision : an application in the framework of GIS.* In proc. of International Conference on the Principles of Knowledge Representation and Reasoning KR'00 (2000).

[WJP 01] Wurbel, E., Jeansoulin, R., Papini, O. : *Spatial information revision : a comparison between 3 approaches.* in proc. of ECSQARU'01, (2001) 454–465.