RÉVISION DES CONNAISSANCES D'UN PROCESSUS DE GÉNÉRALISATION DE DONNÉES GÉOGRAPHIQUES

par Patrick Taillandier

Laboratoire COGIT
Institut géographique national,
2-4 avenue Pasteur 94165 Saint-Mandé Cedex
patrick.taillandier@ign.fr

La généralisation de données géographiques est le processus qui consiste à dériver, à partir de données géographiques détaillées, des données moins détaillées. Nous nous intéressons dans cet article à la révision automatique des connaissances procédurales des systèmes de généralisation, basée sur le paradigme 'agent'. Notre approche consiste à analyser les traces d'exécution du système et d'en extraire, grâce aux techniques d'apprentissage artificiel, de nouvelles connaissances. L'objectif est de rendre le système à la fois plus performant mais aussi capable de s'adapter automatiquement à différentes utilisations et d'évoluer lors de l'ajout de nouveaux éléments. Une première expérimentation sur la généralisation des lotissements a été menée pour valider notre approche.

Mots clés : généralisation de données géographiques, révision des connaissances procédurales, apprentissage artificiel, paradigme agent.

1 Introduction

La généralisation de données géographiques est le processus qui consiste à dériver, à partir de données géographiques détaillées, des données moins détaillées adaptées à un besoin. L'automatisation de ce processus est un problème complexe qui est au centre de nombreuses recherches. Certains de ces travaux proposent de le résoudre par une approche locale et basée sur les connaissances (Brassel et Weibel 1988 ; Beard 1991 ; McMaster et Shea 1992). Le travail présenté dans cet article prend pour cadre les modèles de généralisation automatique basée sur cette approche et utilisant le paradigme 'agent' (Ruas 1999 ; Duchêne 2004).

L'objectif est de proposer des méthodes pour réviser automatiquement, par le biais d'expériences et sans l'intervention d'experts, des connaissances procédurales contenues dans ce type de modèle. L'intérêt de cette révision est d'une part de rendre le système plus efficace (qualité finale du résultat) et plus efficient (vitesse d'obtention du résultat), mais aussi de permettre au système d'évoluer lors de l'ajout de nouveaux éléments (des algorithmes ou de nouvelles mesures) dans le système.

Nous présentons en partie 2, le contexte dans lequel se placent nos travaux. Nous donnons en particulier un bref état de l'art non exhaustif de l'utilisation de l'apprentissage artificiel dans le cadre de la généralisation et nous présentons ensuite le modèle de généralisation AGENT sur lequel est fondée notre approche. La partie 3 est consacrée à la présentation de notre approche. La partie 4 décrit la première expérimentation que nous avons menée ainsi que ses résultats. La partie 5 conclut et présente les perspectives.Une grande partie de cet article est tirée de (Taillandier 2007).

2 Contexte et objectifs

2.1 Généralisation et apprentissage artificiel

L'automatisation du processus de généralisation automatique nécessite l'introduction de nombreuses connaissances qui déterminent le choix des actions à appliquer sur un objet géographique. Ces connaissances peuvent être acquises auprès d'experts. On se retrouve alors confronté au problème du « goulot d'étranglement de l'acquisition des connaissances ». Ce problème a bien été identifié dans le cadre de la généralisation automatique (Rieger et Coulson 1993 ;

Weibel et al. 1995 ; Kilpeläinen 2000). L'une des approches permettant d'y faire face et d'élargir le goulot est de faire appel aux techniques d'apprentissage artificiel. Certains travaux ont déjà utilisé ces techniques dans le cadre de la généralisation (Weibel et al. 1995 ; Mustière 2001 ; Mustière et Ruas 2004). Leur utilisation requiert la résolution de deux types de difficultés (Ruas et al., 2006).

La première difficculté est le choix des connaissances à apprendre et leur formalisation (Mustière et Ruas 2004). Ce choix se révèle capital pour la qualité des résultats que l'on peut obtenir. La seconde concerne la collecte des exemples (Mustière 2001; Ruas et Holzapfel 2003). Il est souvent fastidieux de construire des jeux d'entraînement à partir d'exemples labellisés par des experts. Pour y faire face, une approche possible est d'intégrer directement dans le SIG un environnement facilitant la collecte des exemples ainsi que leur analyse (Duchêne et al. 2005). Une autre approche pour résoudre ce problème est d'acquérir directement les exemples par l'expérimentation sans passer par des experts. On peut citer dans ce domaine deux séries de travaux. Elles prennent pour cadre deux systèmes de généralisation différents, mais qui se basent tous les deux sur l'exploration d'arbres d'états par essais/erreurs en vue de déterminer la meilleure séquence d'actions à appliquer. Ces travaux analysent des généralisations passées pour en déduire des connaissances sur le choix des actions à appliquer pour un état de la carte donné. Burghardt et Neun (2006) utilisent les expériences passées pour construire ce type de connaissances sous la forme d'une base de cas. Dyèvre (2005) et Ruas et al (2006) cherchent pour leur part à acquérir ces connaissances sous forme de règles. L'avantage de ce second type de représentation des connaissances est d'être directement interprétable. Cette dernière approche, dédiée au modèle de généralisation AGENT (Ruas 1999 ; Regnauld 2001), se base sur l'analyse des traces d'exécution pour apprendre de nouvelles règles. Notre approche d'acquisition automatique des connaissances procédurales s'inscrit dans la continuité de cette approche. Elle prend aussi pour cadre le système de généralisation AGENT.

2.2 Le modèle AGENT

Le modèle de généralisation AGENT, initié par (Ruas 1999) et au centre du projet européen AGENT (Lamy et al. 1999 ; Barrault et al. 2001), propose de modéliser les objets géographiques sous la forme d'agents. Ferber (1995, p.14) définit les agents comme « une sorte "d'organisme vivant" dont le comportement qui se résume à communiquer, à agir et, éventuellement, à se reproduire, vise à la satisfaction de ses besoins et de ses objectifs à partir de tous les autres éléments (perceptions, représentations, actions, communications et ressources) dont il dispose ».

- Les agents géographiques gèrent leur propre généralisation en s'auto-appliquant des algorithmes de généralisation (que nous désignerons, dans la suite de cet article, par le terme « actions »). Ce modèle comprend deux types d'agent :
- Les **agents micro** représentent les objets géographiques simples (bâtiments, tronçon de routes, etc.).
- Les **agents méso** représentent les groupes d'objets géographiques. Un agent méso peut aussi bien être composé d'agents micro (un lotissement composé de bâtiments) que d'autres agents méso (un quartier composé de lotissements). Il en résulte une organisation hiérarchique des agents. Les agents de niveau supérieur ont la charge d'activer la généralisation de leur sous-agents.

La généralisation des agents (micro et méso) est guidée par un jeu de contraintes qui traduisent le résultat cartographique souhaité. Un exemple de contrainte est, pour un agent bâtiment, d'être suffisamment gros pour être lisible. Chaque contrainte liée à un agent est modélisée par un objet au sens informatique du terme. Les contraintes ont aussi pour rôle de proposer à leur agent associé, pour chaque état, une liste d'actions à s'appliquer. Par exemple, si la contrainte de taille d'un agent bâtiment perçoit que sa taille est trop petite, elle pourra lui proposer une action de grossissement.

Chaque contrainte porte les attributs suivants :

- La valeur but traduit les spécifications du produit cartographique souhaité.
- La valeur courante traduit l'état courant de la contrainte. Elle est calculée pour chaque état.
- La satisfaction reflète le degré de satisfaction d'une contrainte. Elle est calculée pour chaque état à partir de la valeur courante et de la valeur but. Elle est notée sur 10 (1 = pas du tout satisfaite, 10 = parfaitement satisfaite).

- La priorité reflète l'urgence de traitement d'une contrainte. Plus une contrainte a une priorité élevée, plus ses actions seront appliquées en priorité. Elle est notée sur 5 (1 = pas urgent, 5 = très urgent).
- L'**importance** correspond à l'importance pour le résultat final qu'une contrainte soit satisfaite. Elle est notée sur 5 (1 = pas important, 5 = très important). Cette valeur n'est pas corrélée avec la priorité.

Pour satisfaire au mieux ses contraintes, un agent géographique réalise un cycle d'actions durant lequel il testera les différentes actions proposées par ses contraintes afin d'atteindre le meilleur état possible (fig. 1). La figure 2 donne un exemple d'arbre obtenu pour la généralisation d'un bâtiment. Le passage d'un état à un autre correspond à l'application, par l'agent, d'une action conseillée par au moins l'une de ses contraintes.

3 Approche proposée pour la révision des connaissances

3.1 Approche générale

Tout comme l'approche présentée par (Dyèvre 2005 ; Ruas et al. 2006), notre approche générale est basée sur 3 phases :

- La **phase d'exploration** consiste à tracer le processus pendant qu'il généralise un grand nombre d'objets géographiques. Durant cette phase, le processus utilise les connaissances procédurales contenues initialement dans le système. Les traces contiennent toutes les informations relatives aux succès/échecs des différentes connaissances procédurales du système.
- La **phase d'analyse** consiste à analyser les traces obtenues durant la phase précédente et à en déduire de nouvelles connaissances.
- La **phase d'exploitation** consiste à tester le système avec ces nouvelles connaissances. Si les résultats obtenus ne sont pas suffisamment bons, les deux premières phases peuvent être relancées de façon à obtenir de meilleures connaissances.

3.2 Problématique

Appliquer notre approche à l'acquisition des connaissances procédurales du modèle AGENT, soulève différentes questions. Premièrement, que faut-il tracer pour pouvoir analyser le processus ? On rejoint le problème posé par Mustière et Ruas (2004). Il faut déjà bien définir les connaissances que l'on souhaite acquérir. Dans notre cas, on s'intéresse à toutes les connaissances procédurales ayant trait au parcours de l'arbre d'états. L'objectif est de trouver le plus rapidement possible le meilleur état possible (donc de bien guider l'agent dans ses choix d'actions à appliquer) et de limiter le nombre d'états inutiles (élaguer l'arbre). Il faut dégager les éléments importants qui serviront de base à l'apprentissage pour chacune des connaissances que nous cherchons à réviser. Un point capital qui entre aussi dans les problèmes soulevés par Mustière et Ruas (2004) est la définition du langage de description des exemples et en particulier la caractérisation d'un état de l'arbre d'états. Il faut que le langage de représentation soit assez riche pour qu'il décrive correctement les états et ainsi éviter les bruits de description, mais qu'en même temps il ne soit pas trop riche de façon à éviter l'explosion de la taille de l'espace des hypothèses (ce qui nuirait au résultat de l'apprentissage). Nous caractériserons, dans notre cas, les états par la satisfaction de leurs contraintes. Ces valeurs sont des entiers compris entre 1 et 10 et permettent une caractérisation, correcte à défaut d'être complète, de l'état d'un agent. La qualité de cette caractérisation est directement dépendante des contraintes utilisées. Si certains aspects descriptifs de l'état de l'agent ne sont pas couverts par les contraintes, la qualité des connaissances apprises peut en souffrir.

Une autre question à se poser est celle du choix de la méthode d'apprentissage. Il nous faut en particulier choisir le type d'algorithme à utiliser pour l'apprentissage. Nous sommes dans le cas d'un apprentissage supervisé car nous disposons d'exemples étiquetés, c'est-à-dire d'exemples décrits sous la forme attributs/étiquette, les attributs décrivant les caractéristiques de l'exemple (état de l'agent) et l'étiquette, la classe à laquelle l'exemple appartient (dépend de ce que l'on cherche à apprendre). En effet, l'analyse a posteriori des traces d'exécution du système permet de construire des jeux d'exemples étiquetés. Par ailleurs, nous devons respecter la contrainte de pouvoir faire valider les connaissances apprises par des experts du domaine. Disposer d'un modèle prédictif interprétable devrait, de plus, nous permettre d'apporter des connaissances intéressantes sur la généralisation en elle-même. Cette contrainte nous oblige à faire appel à des techniques d'apprentissage qui permettent de construire un modèle sous forme de règles, ou sous une forme qui puisse être facilement traduite sous forme de règles.

Les techniques les plus simples pour réaliser cet apprentissage sont les techniques d'apprentissage symbolique, telles que les techniques d'induction d'arbres de décision. Il n'existe pas de règles simples définissant le meilleur choix d'algorithme pour un problème d'apprentissage donné. Nous ne chercherons donc pas à tester différents algorithmes, mais nous nous contenterons d'utiliser l'algorithme C4.5 (Quinlan 1993) qui permet d'induire des arbres de décision, facilement traduisibles sous forme de règles. Cet algorithme, très réputé, est résistant au bruit, ce qui le rend adapté à notre problème d'apprentissage.

3.3 Connaissances à réviser

Les connaissances que nous chercherons à réviser peuvent se diviser en deux groupes : celles concernant l'ordre d'application des actions pour un état donné, et celles liées à l'élagage de l'arbre d'états.

3.3.1 Priorité des contraintes

Le modèle AGENT fait intervenir trois facteurs dans l'attribution de l'ordre d'application des actions dans pour un état donné :

1. La priorité des contraintes qui proposent les actions : plus une contrainte a une priorité élevée, plus les actions proposées par celle-ci seront testées en priorité.
2. Si deux contraintes ont la même priorité, le second facteur pris en compte est la satisfaction des contraintes. Les actions proposées par la contrainte la moins satisfaite seront testées en priorité.
3. Le dernier facteur est le poids attribué à chaque action. Lorsqu'une contrainte propose deux actions ou lorsque deux contraintes ont la même priorité et la même satisfaction, l'ordre d'application des actions dépend du poids attribué à chaque action. Plus ce poids sera élevé, plus l'action sera testée en priorité.

Il est donc possible de réviser deux types de connaissances : la priorité de chaque contrainte pour un état donné et le poids de chaque action. Nous ne nous intéresserons dans cet article qu'au premier point. L'approche que nous proposons pour la détermination de la contrainte prioritaire est d'apprendre, pour chaque contrainte, une base de règles définissant pour chaque état si celle-ci doit être prioritaire ou non. Chaque règle de la base se verra attribuer un taux de confiance permettant d'ordonner les contraintes se déclarant comme prioritaires. Les règles apprises sont utilisées comme suit durant la phase d'exploitation : quand un agent est dans un état donné, chaque contrainte vérifie si elle doit ou non se déclarer comme prioritaire. L'agent choisira comme contrainte prioritaire celle dont la règle vérifiée (la règle ayant permis à la contrainte de se déclarer comme prioritaire) a le plus fort taux de confiance.

Exemple

Dans l'exemple suivant, la satisfaction de la contrainte C1 est notée S(C1) et le taux de confiance des règles est noté TC.

Si un agent A possède deux contraintes C1 et C2 qui ont respectivement les bases de règles de priorité suivantes :

- Règles pour C1: si S(C1) > 4 et S(C2) < 3 alors prioritaire avec TC = 80%.
- Règles pour C2: si S(C2) < 10 alors prioritaire avec TC = 75%.

Si A est dans les états suivants :

- État 1 : S(C1) = 7, S(C2) = 8 : Seule C2 se déclare comme prioritaire, donc C2 est prioritaire (ses actions seront donc appliquées en priorité).
- État 2 : S(C1) = 8, S(C2) = 2 : La règle C1 est vérifiée : C1 se déclare comme prioritaire avec TC = 80%. La règle C2 est vérifiée : C2 se déclare comme prioritaire avec TC = 75%. Donc, comme le taux de confiance de C1 est plus élevé que celui de C2, C1 est prioritaire (ses actions seront donc appliquées en priorité).

La construction des jeux d'apprentissage permettant l'induction, par apprentissage artificiel, de ces règles est basée sur l'analyse des arbres d'états obtenus durant la phase d'exploration. On s'intéressera particulièrement aux succès/échecs que chaque action proposée par les contraintes a rencontrés durant la phase d'exploration. Un jeu d'apprentissage est construit par contrainte.

Le principe de construction de jeux d'apprentissage est le suivant : chaque état appartenant au meilleur chemin (séquence d'états débutant à l'état initial et se finissant au meilleur état) est analysé. Si, à partir d'un de ces états, l'action proposée par une contrainte mène à un autre état du meilleur chemin, la contrainte est considérée comme prioritaire. Si l'action proposée mène à un état inutile (état n'appartenant pas au meilleur chemin), elle est considérée comme non prioritaire. La figure 3 donne un exemple de jeux d'apprentissage obtenus.

3.3.2 Élagage de l'arbre d'états

Nous cherchons à élaguer plus fortement l'arbre d'états dans le but d'améliorer l'efficience du processus de généralisation. Deux critères entrent en jeu pour cet élagage : le critère de validité des états et celui de fin de cycle d'actions. Un état invalide est un état à partir duquel le système ne continuera pas l'exploration en profondeur. Mettre fin au cycle d'actions signifie que le système s'arrête de chercher un état meilleur que le meilleur déjà trouvé.

Critère de validité

Nous nous intéresserons d'abord au critère de validité. Le critère de base que nous avons choisi d'intégrer au système de généralisation est qu'un état est valide s'il n'est pas similaire, d'un point de vue satisfaction de chaque contrainte, à un état déjà visité dans l'arbre d'états. Ce critère a été choisi dans le but d'éviter les boucles et d'assurer la convergence du système. Il permet, de plus, d'obtenir des arbres d'états très développés. En effet, il est important pour la qualité des analyses qui peuvent être faites des arbres d'états durant la phase d'exploration que les arbres soient les plus complets possible. Nous allons ajouter à ce critère de base des règles de validité permettant un élagage plus sévère des arbres d'états. Nous chercherons ici à apprendre le concept d'état valide.

De la même manière que pour la priorité des contraintes, nous allons construire, à partir de l'analyse des arbres d'états, un jeu d'apprentissage d'où nous déduirons des règles par apprentissage artificiel. Le principe de construction du jeu d'apprentissage est le suivant : pour chaque arbre d'états, un état sera noté comme valide s'il appartient au meilleur chemin et il sera noté comme invalide s'il n'appartient pas au meilleur chemin alors que son père appartient lui au meilleur chemin. La figure 4 donne un exemple de jeu d'apprentissage obtenu.

Critère d'arrêt du cycle d'actions

Le second critère sur lequel nous allons jouer dans le but d'élaguer nos arbres d'états est le critère de fin de cycle. Le critère de base est que le cycle d'actions se termine lorsqu'un état parfait est obtenu ou lorsque toutes les actions possibles ont été testées pour tous les états. Or, il peut exister certaines configurations où il ne sert plus à rien, pour le système de généralisation, de continuer d'explorer l'arbre d'états car il n'existe pas d'état meilleur que le meilleur déjà trouvé. Nous allons chercher à apprendre ces configurations sous la forme d'états pour lesquels il est inutile de continuer l'exploration de l'arbre d'états. Comme pour les autres connaissances procédurales, nous allons pour cela construire un jeu d'apprentissage et en déduire, par apprentissage artificiel, des règles à partir de ce jeu. Le principe de construction du jeu d'apprentissage est le suivant : pour un arbre d'états donné, tous les états appartenant au meilleur chemin sauf le meilleur seront des états pour lesquels il est nécessaire de continuer à explorer l'arbre d'états. Par contre une fois le meilleur état atteint, le cycle d'actions peut être stoppé, on ne trouvera pas de meilleur état. La figure 4 donne un exemple de jeu d'apprentissage obtenu.

3.4 Conclusion pour le processus de révision des connaissances

La partie 3 a décrit notre approche de révision des connaissances procédurales. Celle-ci est basée sur 3 phases : une phase d'exploration, qui consiste à tracer le processus lorsque celui-ci généralise un grand nombre d'objets géographiques, une phase d'analyse, qui consiste à analyser les traces obtenues pour en déduire des nouvelles connaissances par apprentissage artificiel, et enfin une phase d'exploitation qui consiste à tester le processus avec ses nouvelles connaissances. Trois types de connaissances procédurales ont été apprises durant la phase d'analyse : la priorité des contraintes, le critère de validité des états et le critère de fin de cycle d'actions.

4 Expérimentation : la généralisation des lotissements

4.1 Contexte

Nous avons testé notre approche de révision des connaissances procédurales pour la généralisation cartographique des agents lotissement au 1:50 000. Un agent lotissement est un agent méso composé de divers agents micro (des bâtiments, des routes, des rivières, …) qui doit respecter les conditions suivantes :

- Avoir une densité en bâtiment non forte.
- Contenir au moins deux bâtiments.
- Avoir une surface inférieure à 300m2.
- Etre composé de bâtiments de taille relativement similaire (écart type entre les surfaces des bâtiments inférieur à 70 m2).

L'échelle de référence de nos données initiales est d'environ 1:15 000 (BD TOPO® de l'IGN), l'échelle

souhaitée, le 1:50 000. Nous avons défini quatre contraintes pour nos agents lotissements (fig. 5), auxquelles nous avons attribué des priorités égales. Ainsi, l'ordre d'application des actions dépend uniquement des satisfactions des contraintes proposant les actions. Dans la plupart des cas, ce type de configuration (même priorité pour toutes les contraintes) mène à des résultats corrects à défaut d'être optimaux. La phase d'exploration a conduit à la généralisation de 40 lotissements pris dans les environs de la ville d'Orthez.

4.2 Règles apprises

Nous noterons S(Ci), la satisfaction de la contrainte Ci, et TC le taux de confiance des règles. Les règles obtenues par l'algorithme C4.5 sont décrites ci-dessous.

4.2.1 Règles de choix de la contrainte prioritaire

Nous présentons dans cette partie, les règles apprises pour la priorité des contraintes. Le processus ayant permis d'obtenir ces règles est décrit en partie 3.3.1.

- **Contrainte prox_routes_bâti** :

Règles	TC
SI (S(bâti_satisf) > 7) ALORS **prioritaire**	TC = 92%
SI (S(bâti_satisf) ≤ 7) ET (S(prox_routes_bâti) ≤ 5) ALORS **prioritaire**	TC = 83%
SI (S(bâti_satisf) ≤ 7) ET (S(prox_routes_bâti) > 6) ET (S(prox_bâti) = 8) ALORS **prioritaire**	TC = 71%
SI (S(bâti_satisf) ≤ 7) ET (S(prox_routes_bâti) = 7) ET (S(prox_bâti) > 8) ALORS **prioritaire**	TC = 86%

- **Contrainte bâti_satisf** :

Règles	TC
SI (S(bâti_satisf) ≤ 8) ET (S(prox_routes_bâti) > 4) ET (S(prox_bâti) > 5) ALORS **prioritaire**	TC = 91%
SI (S(bâti_satisf) > 8) ET (S(prox_routes_bâti) > 7) ET (S(prox_bâti) > 5) ALORS **prioritaire**	TC = 93%

- **Contrainte prox_bâti** :

Règles	TC
SI (S(prox_bâti) ≤ 7) ALORS prioritaire	TC = 54%
SI (S(bâti_satisf) > 7) ET (S(prox_bâti) > 7) ALORS prioritaire	TC = 67%

D'après ces règles, l'action proposée par la contrainte bâti_satisf sera appliquée en priorité dans la plupart des cas. Cette action consiste à déclencher la généralisation des bâtiments composant le lotissement. Dans la très grande majorité des cas, cette action est appliquée qu'une seule fois par chemin de l'arbre d'états. On peut noter que le fait de privilégier la contrainte de satisfaction des bâtiments dans le cas d'un groupe non dense de bâtiments comme nos lotissements avait déjà été remarqué par (Duchêne 2004). Le seul cas où l'action proposée par la contrainte bâti_satisf ne sera pas appliquée en priorité est le cas où les bâtiments sont déjà bien généralisés et où il y a peu de conflits de proximité entre bâtiments. Dans ce cas, les actions de la contrainte prox_routes_bâti seront appliquées en priorité.

Une fois le problème de généralisation des bâtiments résolu, on passera dans la plupart des cas à une phase de déplacement et de suppression de bâtiments (actions proposées par la contrainte prox_routes_bâti). En effet, la généralisation au 1:50 000 des bâtiments entraîne généralement leur grossissement, ce qui a pour conséquence l'introduction de superpositions entre bâtiments et entre bâtiments et routes. Il est donc souvent nécessaire d'avoir recours à des actions permettant de résoudre ce problème.

On peut remarquer que les actions proposées par la contrainte prox_bâti (des agrégations) peuvent aussi aider à résoudre ce problème, mais qu'en général le système lui préfère largement les actions proposées par la contrainte prox_routes_bâti.

4.2.2 Critère de validité des états

Nous présentons dans cette partie, la règle de validité apprise. Le processus ayant permis d'obtenir cette règle est décrit en partie 3.3.2. La règle obtenue est la suivante :

Règles
SI (S(bâti_satisf) ≤ 9) ET (S(nb_bâti) ≤ 8) ALORS **état invalide**

Le seul cas où un état est considéré comme invalide pour cette règle est le cas où les bâtiments ne sont pas encore généralisés (ou très mal généralisés) et où, en même temps, trop de bâtiments ont déjà été supprimés. Cette règle semble pertinente. En effet, généraliser les bâtiments peut signifier l'apparition de conflits de superposition et de proximité (car, dans la plupart des cas, les bâtiments vont augmenter de taille après généralisation) et le système ne peut plus se permettre de supprimer des bâtiments au risque de voir la satisfaction de la contrainte nb_bâti s'effondrer encore plus.

4.2.3 Critère de fin de cycle d'actions

Nous présentons dans cette partie, la règle de fin de cycle apprise. Le processus ayant permis d'obtenir cette règle est décrit en partie 3.3.2. La règle obtenue est la suivante :

Règles
SI (S(bâti_satisf) = 10) ET (S(prox_routes_bâti) = 10) ALORS **fin du cycle d'actions**

D'après cette règle, le système considère qu'il est inutile de continuer à explorer l'arbre d'états si les contraintes bâti_satisf et prox_routes_bâti sont parfaitement satisfaites. Pour justifier cela, nous pouvons noter que ces deux contraintes sont les seules à proposer des actions réellement efficaces. Une fois que leur satisfaction maximale est atteinte, les contraintes ne proposent plus d'actions, donc dans notre cas, plus aucune autre action réellement efficace n'est proposée une fois que la condition de cette règle est vérifiée.

4.3 Test des connaissances apprises

Le processus de généralisation a été testé avec les connaissances initiales (avec les critères basiques de validité des états et de fin de cycle décrits dans la partie 3.3.2) puis avec les nouvelles connaissances sur 40 lotissements pris dans une zone différente de celle utilisée durant la phase d'exploration. Le critère de validité basique est qu'un état est valide s'il n'est pas exactement similaire, d'un point de vue de la satisfaction de ses contraintes, à aucun état déjà visité dans l'arbre d'états. Le critère de fin de cycle basique est que le cycle d'actions se termine lorsqu'un état parfait est atteint (ou que toutes les actions possibles ont été testées pour tous les états).

Les résultats (fig. 8) montrent une amélioration du point de vue de l'efficacité (satisfaction moyenne obtenue) mais aussi du point de vue de l'efficience (nombre moyen d'états explorés). Le nombre moyen d'états visités par lotissement a été divisé par 6 avec nos nouvelles connaissances, alors que la qualité cartographique des résultats a légèrement progressé. La satisfaction globale des lotissements a en effet très légèrement augmenté et on peut observer, sur la figure 7, que les connaissances apprises ont permis d'éviter quelques problèmes de gros agglomérats et de proximité présents dans les connaissances initiales. Le temps moyen nécessaire à la généralisation d'un lotissement a lui aussi diminué d'un facteur comparable à celui du nombre d'états. La perte de temps due à l'évaluation des règles est totalement négligeable comparée au temps nécessaire à l'application des actions de généralisation.

Cette expérimentation démontre qu'il est possible d'améliorer significativement la qualité des connaissances procédurales incluses dans ce système de généralisation par un processus introspectif. Elle nous permet donc de valider notre approche générale.

5 Conclusion et perspectives

Nous avons souligné, dans cet article, l'intérêt que pouvait avoir l'intégration d'un module de révision automatique des connaissances procédurales dans un système de généralisation à base de connaissances. Nous avons proposé une approche de révision des connaissances procédurales basée sur l'introspection et sur des techniques d'apprentissage artificiel. Dans le cadre de l'utilisation du système de généralisation automatique AGENT, nous avons développé différentes méthodes et algorithmes permettant à la fois d'acquérir des connaissances sur l'ordre d'application des actions de généralisation, mais aussi sur l'élagage des arbres d'états construits. Une expérimentation sur la généralisation des lotissements au 1:50 000 a permis de valider notre approche et de prouver la qualité des connaissances qu'elle nous permet d'appréhender.

Parmi les connaissances du modèle AGENT, que nous n'avons pas cherché à acquérir dans le travail

présenté dans cet article, figurent les connaissances propres au choix des actions proposées par les contraintes ainsi que leur poids associé. Des méthodes permettant à chaque contrainte de définir, pour un état donné de l'agent, les actions les plus appropriées à proposer sont en cours de développement.

Par ailleurs, nous nous sommes, pour l'instant, contentés d'acquérir de nouvelles connaissances que nous avons substituées aux anciennes. Nos futurs travaux porteront sur la mise en place de méthodes permettant de mieux prendre en compte les connaissances initiales dans le processus de révision. Développer de telles méthodes nécessite au préalable de disposer d'un formalisme commun d'expression des connaissances ainsi que de structures de stockage permettant de les gérer de façon totalement dynamique. Un travail de modélisation des connaissances doit donc être entrepris.

Bibliographie

BARRAULT M., REGNAULD N., DUCHENE C., HAIRE K.,, BAEIJS C., DEMAZEAU Y., HARDY P., MACKANESS W., RUAS A., WEIBEL R., 2001,. "Integrating multi-agent, object-oriented, and algorithmic techniques for improved automated map generalisation", dans *20th ICC conference*, vol. 3. p. 2100-2116.

BEARD K., 1991, *Constraints on rules formation, map generalization*, Buttenfield B. et McMaster R., ed,, Longman, p. 121-135.

BRASSEL K., WEIBEL R., 1988, "A review and conceptual framework of automated map generalization" ,*International Journal of Geographical Information Systems*. vol. 2, n°3, p. 229-244.

BURGHARDT D., NEUN M., 2006, Automated sequencing of generalisation services based on collaborative filtering, dans *4th International Conference GIScience*.

DUCHÊNE C, 2004, *Généralisation cartographique par agents communicants : le modèle CartACom,*, thèse de doctorat, Université Paris VI et laboratoire COGIT.

DUCHENE C., DADOU D., RUAS A., 2005, "Helping the capture of expert knowledge to support generalisation"., dans *ICA Workshop on generalization and multiple representation*, La Corona, Spain.

DYEVRE A., 2005, *Analyse d'un processus de généralisation cartographique à l'aide d'apprentissage artificiel*, rapport de Master 2, Université Paris IV et Laboratoire COGIT.

FERBER J., 1995, *Les systèmes multi-agents - vers une intelligence collective*, Paris : InterEditions.

KILPELÄINEN T., 2000, "Knowledge Acquisition for Generalisation Rules", *Cartography and Geographic Information Science*, vol. 27, n°1, p. 41-50.

MCMASTER R.B., SHEA K.S., 1992, *Generalization in Digital Cartography*, Washington, Association of American Geographers.

MUSTIÈRE S., 2001, *Apprentissage supervisé pour la généralisation cartographique.*, thèse de doctorat, Université Paris VI, laboratoire COGIT.

MUSTIÈRE S., DUCHÊNE C., 2001, "Comparison of different approaches to combine road generalisation algorithms: GALBE, AGENT and CartoLearn", dans *4th ICA Workshop on generalisation*.

MUSTIÈRE S., RUAS A.., 2004, « Vers une réconciliation des experts et des systèmes - Expériences d'utilisation de méthodes d'apprentissage artificiel pour la généralisation et l'intégration des bases de données géographiques. », dans *Actes des Journées Cassini*.

QUINLAN J. R., 1993, *C4.5 Programs for Machine Learning,* San Mateo, CA: Morgan Kaufmann.

REGNAULD N., 2001, "Constraint based mechanism to achieve automatic generalisation using an agent modelling", dans *9 GISRUK conference*

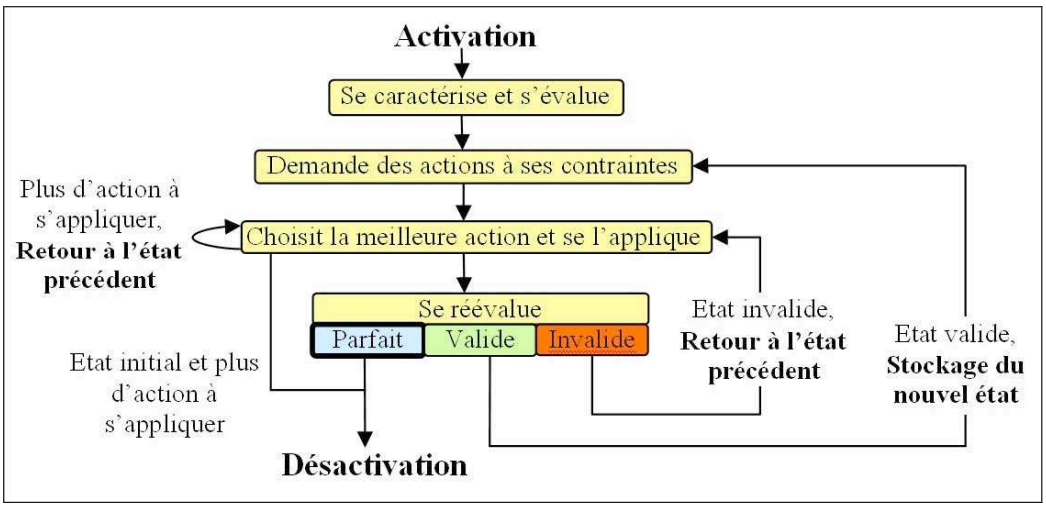

Figure 1 : Vue simplifiée du cycle d'actions d'un agent

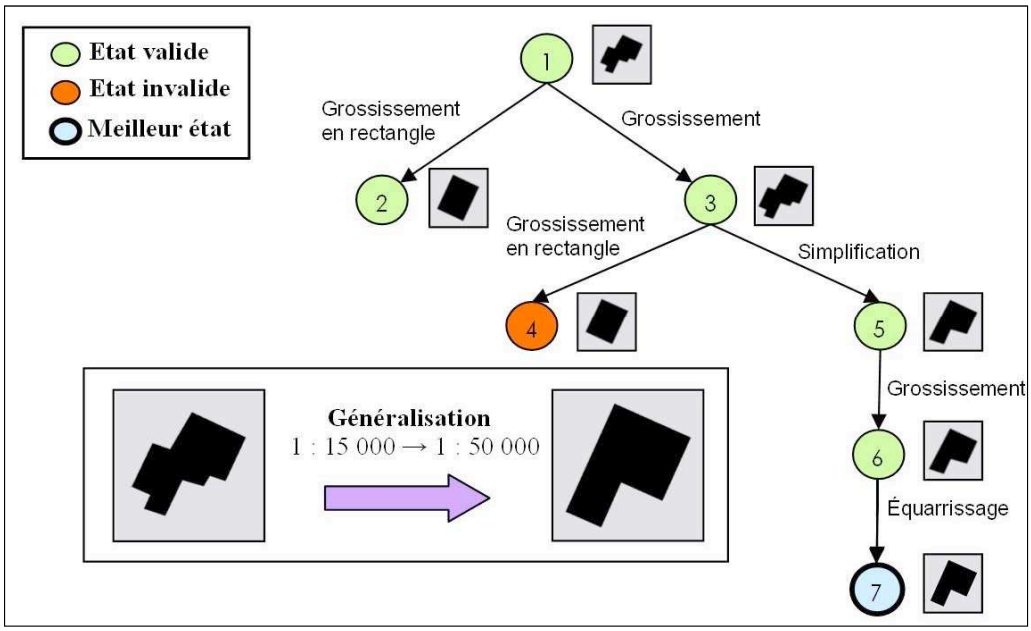

Figure 2 : Exemple d'arbre d'états pour un bâtiment

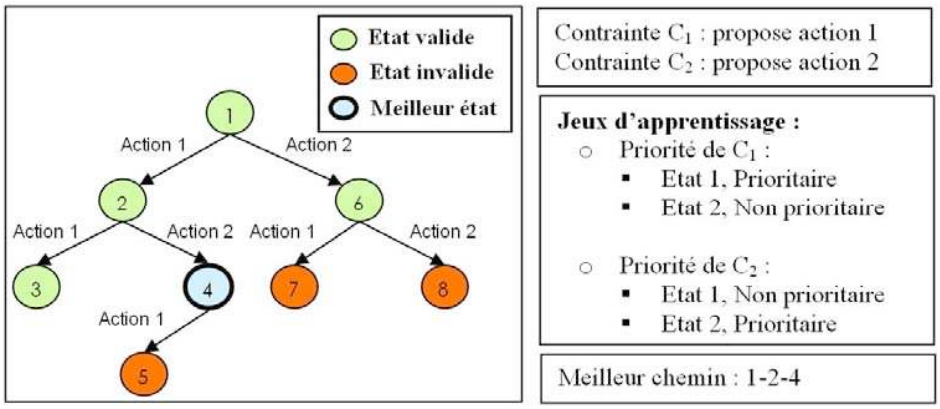

Figure 3 : Exemple de jeux d'apprentissage pour la priorité des contraintes

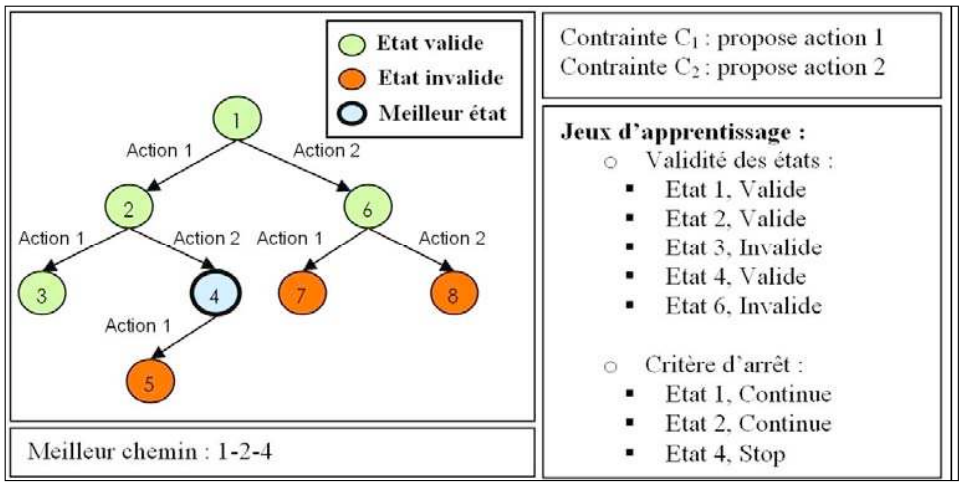

Figure 4 : Exemple d'arbre d'états pour un bâtiment

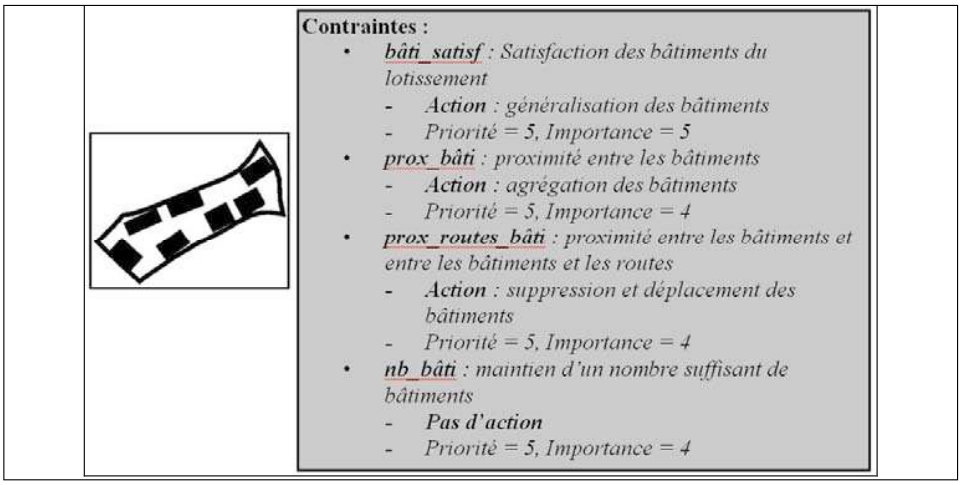

Figure 5 : Contraintes des agents lotissements

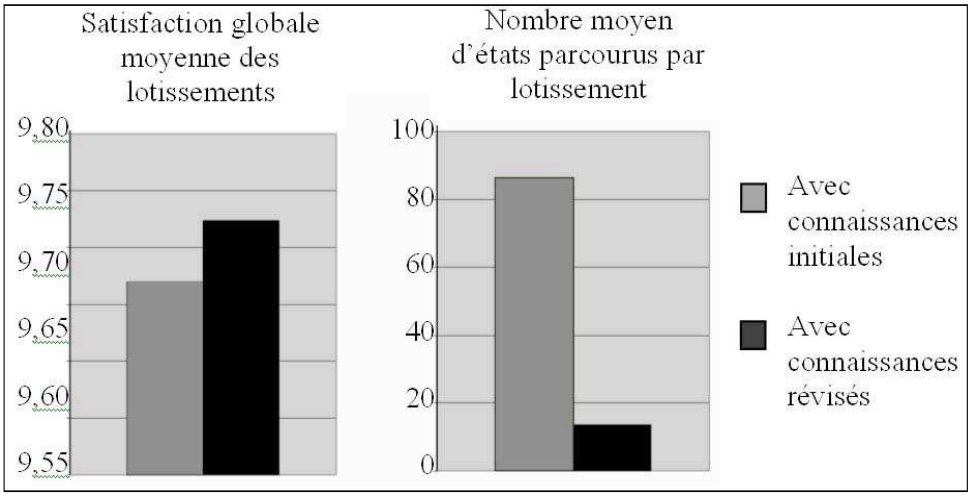

Figure 6 : Exemple d'arbre d'états pour un bâtiment

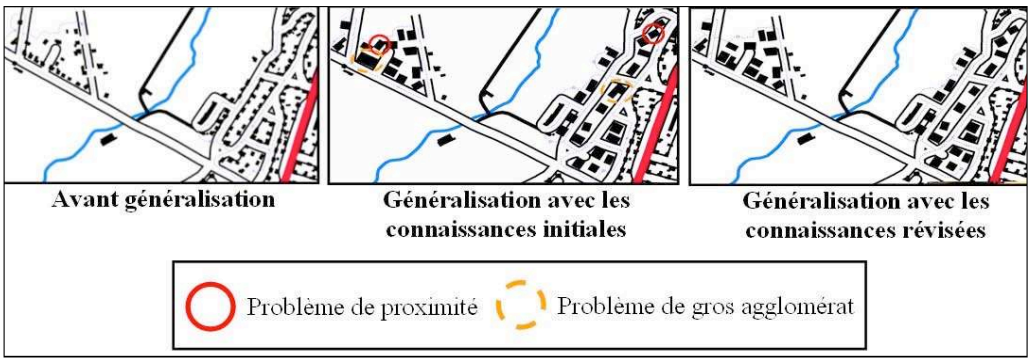

Figure 7 : Exemple de résultats obtenus