



HAL
open science

Adapter des cas en utilisant un opérateur de révision ou des règles

Julien Cojan, Jean Lieber

► **To cite this version:**

Julien Cojan, Jean Lieber. Adapter des cas en utilisant un opérateur de révision ou des règles. Journée Intelligence Artificielle Fondamentale, Laurence Cholvy, Sébastien Konieczny, Jun 2010, Strasbourg, France. inria-00512529

HAL Id: inria-00512529

<https://inria.hal.science/inria-00512529v1>

Submitted on 30 Aug 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Adapter des cas en utilisant un opérateur de révision ou des règles

Julien Cojan et Jean Lieber

UHP-Nancy 1 – LORIA (UMR 7503 CNRS-INPL-INRIA-Nancy 2-UHP)
BP 239, 54506 Vandœuvre-lès-Nancy, France
{Julien.Cojan, Jean.Lieber}@loria.fr

Résumé : L'étape d'adaptation d'un système de raisonnement à partir de cas consiste à modifier un cas source en vue de la résolution d'un problème cible. Une des approches de l'adaptation s'appuie sur un opérateur de révision : la connaissance associée au cas source est révisée par les contraintes du problème cible, en tenant compte des connaissances du domaine. Une autre approche de l'adaptation consiste à appliquer des règles d'adaptation. L'objectif de cet article est d'étudier les liens entre ces deux approches de l'adaptation. Étant donné un ensemble de règles d'adaptation, on peut construire un opérateur de révision fondé sur ces règles qui fait coïncider les deux approches. On peut montrer également comment l'approche utilisant la révision peut étendre l'approche à partir de règles. Cela ne rend pas les travaux sur l'approche à base de règles d'adaptation inutiles. Au contraire, l'acquisition de telles règles dans un domaine particulier permet de définir un opérateur de révision approprié pour l'adaptation dans ce domaine.

Mots-clés : raisonnement à partir de cas, adaptation, révision, règles d'adaptation.

1 Introduction

Le raisonnement à partir de cas (RÀPC, Riesbeck & Schank (1989)) est un mode de résolution de problèmes s'appuyant sur la réutilisation d'expériences antérieures, appelées cas. L'ensemble (fini) des cas est la base de cas et un cas de cette base est appelé *cas source*. Le problème à résoudre lors d'une session de RÀPC est appelé *problème cible*. On distingue souvent deux grandes étapes d'une session de RÀPC : la remémoration, qui consiste à sélectionner un cas source jugé similaire au problème cible, et l'adaptation de ce cas source, dont l'objectif est sa réutilisation en vue de la résolution du problème cible.

Le RÀPC est abordé de façon différente selon la complétude supposée ou non des connaissances de résolution de problèmes du domaine considéré. Si on suppose que ces connaissances, pour une application donnée, sont complètes, autrement dit qu'on dispose d'une inférence déductive permettant de décider si une solution donnée résout un problème donné, alors, l'objectif du RÀPC est de gagner du temps de calcul par rapport à une résolution directe. C'est typiquement le cas pour la planification à partir de cas dans les domaines dépourvus d'incertitudes (voir, p. ex., Veloso (1994)). À l'inverse,

si les connaissances du domaine sont incomplètes, le RÀPC a pour but de *proposer* une solution à un problème, sans qu'il soit certain que *cette* solution résout effectivement *ce* problème. Cette distinction a été baptisée « analogie heuristique » *versus* « analogie recours » dans Coulon *et al.* (1990). On pourrait envisager de la rebaptiser « RÀPC déductif » *versus* « RÀPC hypothétique ».

Ce travail se situe dans le cadre de l'analogie recours et, plus particulièrement, dans le cadre de l'adaptation, qui est, relativement à la remémoration, peu étudiée dans la littérature du RÀPC. Il existe néanmoins plusieurs approches générales de l'adaptation et l'objet de cet article est la comparaison de deux de ces approches : l'adaptation s'appuyant sur un opérateur de révision et l'adaptation à base de règles.

Afin de comparer ces deux approches, nous nous situons dans un cadre formel décrit à la section 2. Puis, nous décrivons successivement ces approches aux sections 3 et 4. La section 5 étudie les liens entre ces deux approches de l'adaptation et conduit à une approche combinant révision et utilisation de règles. La section 6 conclut et présente des perspectives de ce travail.

Remarque 1

La description et la comparaison des approches de l'adaptation présentées dans cet article sont envisagées essentiellement du point de vue des inférences qu'elles permettent de réaliser, pas du point de vue de leurs complexités ni même de leurs décidabilités.

2 Préliminaires

Beaucoup de travaux sur la révision s'appuient sur un formalisme propositionnel avec un nombre fini de variables (voir, p. ex., Katsuno & Mendelzon (1991)). Beaucoup de travaux sur le RÀPC s'appuient sur une représentation de type « attribut-valeur simple » (une valeur simple étant un booléen, un entier, un réel ou une valeur d'un type énuméré). Le formalisme utilisé dans cet article a été choisi parce qu'il recouvre ces deux types de représentation.

On se donne un ensemble \mathcal{U} dont les éléments représentent des expériences individuelles (i.e., des expériences « élémentaires », complètement « instanciées »). Dans un cadre propositionnel, \mathcal{U} sera l'ensemble des interprétations. Dans un cadre attribut-valeur simple, \mathcal{U} sera l'ensemble des n -uplets (x_1, \dots, x_n) où x_i est une valeur du type de l'attribut numéro i .

De façon générale, un cas \mathcal{C} représente un ensemble de situations individuelles : $\mathcal{C} \subseteq \mathcal{U}$. Nous ne précisons pas davantage le langage des cas : il faut simplement qu'il soit récursif et que les cas construits par les opérateurs présentés dans cet article appartiennent à ce langage. Dans un cadre propositionnel, \mathcal{C} sera une formule, assimilée à l'ensemble de ses modèles.

Lors d'une adaptation, on a un cas source, noté *Source*, et un problème cible. Celui-ci, de façon générale, est représenté par un cas, noté *Cible* (le cas cible). Dans beaucoup d'applications, un cas source correspond à une expérience individuelle $x \in \mathcal{U}$. Dans ce cas, *Source* = $\{x\}$ et on parle de cas spécifique. En revanche, le problème cible ne spécifie que « la partie problème » du cas. La distinction entre « partie problème » et « partie solution » d'un cas est inutile pour l'adaptation par révision (sec-

tion 3) mais nous la ferons et la détaillerons pour l'adaptation par règles (section 4). Un système de RÀPC s'appuie généralement sur CD, une base de connaissances du domaine (aussi appelée théorie — ou ontologie — du domaine). CD donne des conditions nécessaires pour qu'un $x \in \mathcal{U}$ soit licite. On l'assimilera à un sous-ensemble de \mathcal{U} : si x est licite, alors $x \in \mathcal{U}$. Les cas sont supposés cohérents avec les connaissances du domaine, autrement dit $CD \cap \text{Source} \neq \emptyset$ et $CD \cap \text{Cible} \neq \emptyset$. De façon générale, un cas C doit être appréhendé en accord avec les connaissances du domaine ; autrement dit, on considère que deux cas sont équivalents si leurs intersections avec CD donne le même résultat.

Le but de l'adaptation est d'apporter des informations sur Cible, autrement dit à inférer un cas, noté CibleRésolue, tel que $CD \cap \text{CibleRésolue} \subseteq CD \cap \text{Cible}$.

Une application jouet.

L'exemple suivant est donné à titre purement illustratif des différentes notions introduites dans cet article. L'objectif de l'application est de calculer une valeur approchée d'un logarithme décimal en s'appuyant sur une table de logarithmes qui constituera la base de cas. Une ligne de cette table est un couple $(x_1, x_2) \in \mathcal{U}$ où $\mathcal{U} = \mathbb{R}_+^* \times \mathbb{R}$ ($\mathbb{R}_+^* =]0; +\infty[$) et où $x_2 \simeq \log x_1$ à $\varepsilon > 0$ fixé près. Un cas source est donc spécifique et s'écrit $\text{Source} = \{(x_1, x_2)\}$.

Un problème cible est un $c \in \mathbb{R}_+^*$ dont on veut une valeur approchée du logarithme. Il est représenté par le cas $\text{Cible} = \{c\} \times \mathbb{R}$.

Les connaissances générales qu'on souhaite utiliser sont les suivantes (pour $t \in \mathbb{R}_+^*$) :

$$\log t \leq 0,44(t - 1) \quad (1)$$

$$\log 1/t = -\log t \quad (2)$$

$$\log 10t = \log t + 1 \quad (3)$$

On se place dans un cadre d'analogie recours : l'approche cherchée proposera une valeur approchée de $\log c$ sans qu'on contrôle l'erreur commise.

3 Adapter en utilisant un opérateur de révision

Dans un formalisme donné, un opérateur de révision $\dot{+}$ associe à deux bases de connaissances ψ et μ une base de connaissances $\psi \dot{+} \mu$, révision de ψ par μ . Intuitivement, $\psi \dot{+} \mu$ est obtenue en faisant un changement minimal de ψ en ψ' , afin que la conjonction de ψ' et μ soit consistante. $\psi \dot{+} \mu$ sera alors cette conjonction. La notion de changement minimal peut être modélisée de plusieurs façons différentes, ce qui fait qu'il n'y a pas unicité de l'opérateur de révision. Néanmoins, des postulats ont été proposés pour un tel opérateur, les plus connus étant les postulats AGM (Alchourrón *et al.* (1985)).

Ces postulats ont été appliqués à la logique propositionnelle (Katsuno & Mendelzon (1991)) et beaucoup étudiés dans le cadre de ce formalisme (voir, p. ex. Dalal (1988)). En particulier, si on se donne une distance sur l'ensemble \mathcal{U} des interprétations, on définit l'opérateur $\dot{+}_d$ ainsi : l'ensemble des modèles de $\psi \dot{+}_d \mu$ est l'ensemble des

modèles de μ à distance minimale de l'ensemble des modèles de ψ . On peut généraliser cette idée en définissant un opérateur de révision $\dot{+}_d$ sur les sous-ensembles d'un espace métrique (\mathcal{U}, d) :

$$\mathcal{C} \dot{+}_d \mathcal{D} = \{y \in \mathcal{D} \mid d(\mathcal{C}, y) = d(\mathcal{C}, \mathcal{D})\}$$

avec $d(\mathcal{C}, y) = \inf_{x \in \mathcal{C}} d(x, y)$ et $d(\mathcal{C}, \mathcal{D}) = \inf_{x \in \mathcal{C}, y \in \mathcal{D}} d(x, y)$

Le minimum est remplacé par une borne inférieure, puisque le minimum n'est pas nécessairement atteint.

$\dot{+}_d$ ne vérifie pas tout à fait les postulats AGM, précisément pour cette raison. Par exemple, si $\mathcal{U} = \mathbb{R}$ et $d : (x, y) \in \mathbb{R}^2 \mapsto |y - x|$, alors $[0; 1] \dot{+}_d [2; 3] = \emptyset$, alors que les postulats AGM imposent que $\psi \dot{+} \mu$ soit satisfiable dès lors que μ l'est¹. Cet opérateur de révision est une généralisation immédiate de l'opérateur de Dalal (1988).

Il n'est pas nécessaire que d vérifie toutes les propriétés des distances. Il suffit que d soit à valeur dans $[0; +\infty]$ et vérifie l'axiome de séparation². Dans la suite de l'article, une « distance » est supposée avoir ces propriétés (aucune autre propriété n'est requise, en particulier, pas la symétrie).

Étant donné un opérateur de révision $\dot{+}$, la $\dot{+}$ -adaptation consiste simplement à utiliser l'opérateur de révision pour effectuer l'adaptation, en tenant compte des connaissances du domaine :

$$\text{CibleRésolue} = (\text{CD} \cap \text{Source}) \dot{+} (\text{CD} \cap \text{Cible}) \quad (4)$$

L'idée intuitive est la réutilisation maximale (le changement minimal) du cas source, en tenant compte des contraintes imposées par le problème cible. La première propriété attendue du résultat d'une adaptation est qu'elle soit cohérente avec les informations associées aux connaissances du domaine et au problème cible, autrement dit qu'on ait

$$\text{CibleRésolue} \subseteq \text{CD} \cap \text{Cible}$$

Cela est assuré par la propriété sur $\dot{+}$ suivante : $\mathcal{C} \dot{+} \mathcal{D} \subseteq \mathcal{D}$ (en logique propositionnelle : $\psi \dot{+} \mu \models \mu$). C'est une propriété vérifiée par un opérateur respectant les postulats AGM et par un opérateur $\dot{+}_d$ tel que défini ci-dessus.

Dans Lieber (2007), cette approche de l'adaptation est étudiée dans le cadre de la logique propositionnelle et est qualifiée d'adaptation « conservatrice », puisqu'elle s'appuie sur l'idée de conserver au maximum les caractéristiques du cas source. Néanmoins, nous avons abandonné cet adjectif car nous présumons que la grande majorité (la totalité ?) des approches de l'adaptation s'inscrivant dans le cadre de l'analogie recours s'appuient sur ce principe et peuvent s'exprimer par une $\dot{+}$ -adaptation. C'est ce que nous montrons plus loin pour l'approche de l'adaptation par règles. Dans Cojan & Lieber (2008), la $\dot{+}$ -adaptation est étudiée dans un cadre formel similaire à celui de cet

¹Néanmoins, au prix de la modification suivante des postulats, $\dot{+}_d$ vérifie ces postulats modifiés : on ajoute la condition selon laquelle les sous-ensembles concernés par les postulats vérifient le fait que si \mathcal{C} et \mathcal{D} sont deux tels sous-ensembles, alors il existe $x \in \mathcal{C}$ et $y \in \mathcal{D}$ tels que $d(\mathcal{C}, \mathcal{D}) = d(x, y)$ (preuve dans Cojan & Lieber (2008) ; pour $\mathcal{U} = \mathbb{R}^n$ et une distance d issue d'une norme, il suffit que \mathcal{C} et \mathcal{D} soient des compacts).

²À savoir $d(x, y) = 0$ ssi $x = y$ pour $x, y \in \mathcal{U}$. Bien évidemment, la fonction qui à $\mathcal{C}, \mathcal{D} \subseteq \mathcal{U}$ associe $d(\mathcal{C}, \mathcal{D})$ ne vérifie pas cet axiome : $d(\mathcal{C}, \mathcal{D})$ n'est qu'une notation pratique.

article. Dans Cojan & Lieber (2009), elle est étendue au problème de la combinaison des cas. Dans Cojan & Lieber (2010), une approche similaire de l'adaptation est décrite dans la logique de descriptions \mathcal{ALC} .

Une \dagger -adaptation pour l'application jouet.

Pour mettre en œuvre la \dagger -adaptation, il faut déterminer un opérateur \dagger sur \mathcal{U} et les connaissances du domaine. En l'absence (provisoire) de toute contrainte sur \dagger , on fait le choix de \dagger_d où d est une distance « simple » sur \mathcal{U} . En l'occurrence, nous choisissons

$$d : ((x_1, x_2), (y_1, y_2)) \in \mathcal{U}^2 \mapsto |y_1 - x_1| + |y_2 - x_2|$$

Les connaissances du domaine utilisées sont issues de l'équation (1), qui se traduit par

$$CD = \{(x_1, x_2) \mid x_2 \leq 0,44(x_1 - 1)\}$$

Avec ces choix, la \dagger_d -adaptation consiste simplement en une copie-correction : on copie la valeur associée à la source qu'on rectifie éventuellement *a minima* pour rétablir la cohérence avec les connaissances du domaine :

si Source = $\{(x_1, y_1)\}$ et Cible = $\{x_2\} \times \mathbb{R}$
alors CibleRésolue = $\{(x_2, y_2)\}$ avec $y_2 = \min(y_1, 0,44(x_2 - 1))$

En l'occurrence, la \dagger -adaptation donne un résultat quelque peu décevant (« Tout ça pour ça ! »). Le coupable est le choix de d , comme nous le verrons plus loin.

4 Adapter en utilisant des règles d'adaptation

Pour l'approche de l'adaptation présentée dans cette section, deux hypothèses sont faites : d'une part chaque cas est représenté en distinguant sa partie problème et sa partie solution, d'autre part les cas sources sont des cas spécifiques.

La distinction entre problèmes et solutions signifie d'abord que \mathcal{U} s'écrit $\mathcal{U}_{pb} \times \mathcal{U}_{sol}$ où un élément de \mathcal{U}_{pb} représente une instance de problème et un élément de \mathcal{U}_{sol} représente une instance de solution. Ensuite, cette distinction signifie que tout cas C peut s'écrire $P \times S$ où P (resp. S) appartient au langage des problèmes (resp., des solutions), qui est un sous-ensemble récursif de $2^{\mathcal{U}_{pb}}$ (resp., de $2^{\mathcal{U}_{sol}}$). Pour le cas Cible, sa partie problème est parfaitement spécifiée (on résout des problèmes cibles spécifiques) et est donc de la forme $\{cible\}$ où $cible \in \mathcal{U}_{sol}$, et sa partie solution est non spécifiée, c'est donc \mathcal{U}_{sol} . Donc Cible = $\{cible\} \times \mathcal{U}_{sol}$. Notons que la majorité des systèmes de RÀPC font cette distinction problème-solution *a priori*. Cependant, quelques systèmes « découpent » à chaque adaptation le cas source en une partie qui s'apparie avec l'instance cible (qui devient la partie problème) et une autre partie (la partie solution, à adapter). C'est le cas par exemple du système REBECAS (Rougegrez (1994)).

La spécificité d'un cas Source se traduit par le fait qu'il s'écrit Source = $\{x\}$ où $x \in \mathcal{U} = \mathcal{U}_{pb} \times \mathcal{U}_{sol}$. On notera $x = (srce, sol(srce))$, i.e., Source = $\{srce\} \times \{sol(srce)\}$. La résolution du problème cible consiste à spécifier sa solution, i.e.,

à trouver $\text{sol}(\text{cible}) \in \mathcal{U}_{\text{sol}} : \text{CibleRésolue} = \{\text{cible}\} \times \{\text{sol}(\text{cible})\} = \{(\text{cible}, \text{sol}(\text{cible}))\}$.

Un problème d'adaptation est la donnée d'un triplet $(\text{srce}, \text{sol}(\text{srce}), \text{cible}) \in \mathcal{U}_{\text{pb}} \times \mathcal{U}_{\text{sol}} \times \mathcal{U}_{\text{pb}}$ où $\{(\text{srce}, \text{sol}(\text{srce}))\}$ est le cas source à adapter et $\{\text{cible}\}$ est le problème cible. De façon générale, une règle d'adaptation s'applique sur un problème d'adaptation vérifiant certaines conditions et produit une instance de solution $\text{sol}(\text{cible})$ de cible (ou, pour être plus précis, une proposition d'instance de solution de cible, puisque nous sommes dans un cadre d'analogie recours). Dans cet article, nous nous limitons aux règles d'adaptation appelées *reformulations* dans Melis *et al.* (1998). Une telle règle est un couple (r, \mathcal{A}_r) où r est une relation entre problèmes et où \mathcal{A}_r est une fonction qui à un problème d'adaptation associe une solution. Cette règle d'adaptation s'interprète de la façon suivante :

si $\text{srce } r \text{ cible}$
alors $\text{sol}(\text{cible}) = \mathcal{A}_r(\text{srce}, \text{sol}(\text{srce}), \text{cible})$ résout probablement cible (5)

On note RA l'ensemble fini des règles d'adaptation du système de RÀPC considéré.

On peut composer les règles d'adaptation de la façon suivante. On appelle *chemin de similarité* de l'instance de problème srce à l'instance de problème cible , un chemin reliant ces deux instances de problème par une composition de relations r telles que $(r, \mathcal{A}_r) \in \text{RA}$. On peut l'écrire $\{(\text{pb}_{i-1}, r_i, \text{pb}_i)\}_{1 \leq i \leq q}$ où $\text{pb}_0 = \text{srce}$, $\text{pb}_q = \text{cible}$ et $\text{pb}_{i-1} r_i \text{ pb}_i$ pour tout i . À srce et cible donnés, il peut y avoir plusieurs chemins de similarité. Pour choisir entre eux, on utilise une fonction de coût qui à $(r, \mathcal{A}_r) \in \text{RA}$ associe $\text{coût}(r) > 0$. Le coût d'un chemin de similarité se calcule de façon additive, c'est $\sum_i \text{coût}(r_i)$. Pour fixer les idées, on peut définir $\text{coût}(r) = -\log P_r$ où P_r est une estimation de la probabilité que l'application de (5) résout effectivement cible. L'additivité du coût traduit alors une hypothèse d'indépendance.

Dans le cas général, il peut y avoir plusieurs chemins de similarité de coût minimal de srce à cible , conduisant à plusieurs instances de solutions de cible . On note $\text{Solutions}(\text{cible})$ cet ensemble de solutions. Si le système n'a pas de connaissances annexes pour choisir une instance de solution dans $\text{Solutions}(\text{cible})$, alors le choix est effectué par l'utilisateur (ou laissé au hasard).

L'adaptation consiste d'abord à construire le chemin de similarité de moindre coût, puis à « suivre » ce chemin, en appliquant successivement les \mathcal{A}_{r_i} . L'étape de construction du chemin de similarité est un problème algorithmique qui dépend de la forme des r_i et qui peut se révéler complexe ; dans certains cas, il est indécidable (car la composition de deux relations décidables n'est pas nécessairement décidable). Dans Lieber (2002), un algorithme établissant un chemin de similarité durant la remémoration, qui combine un parcours de hiérarchie avec une recherche A* et qui a été implanté pour plusieurs applications, est présenté.

Étant donné un chemin de similarité $cs = \{(\text{pb}_{i-1}, r_i, \text{pb}_i)\}_{1 \leq i \leq q}$ de $\text{srce} \in \mathcal{U}_{\text{pb}}$ à $\text{cible} \in \mathcal{U}_{\text{pb}}$ et $\text{sol}(\text{srce}) \in \mathcal{U}_{\text{sol}}$, soit $\text{sol}(\text{pb}_i) = \mathcal{A}_{r_i}(\text{pb}_{i-1}, \text{sol}(\text{pb}_{i-1}), \text{sol}(\text{pb}_i))$ pour $1 \leq i \leq q$. $\text{sol}(\text{pb}_q) = \text{sol}(\text{cible})$ est une proposition d'instance de solution pour cible. On appelle *chemin d'adaptation* de $(\text{srce}, \text{sol}(\text{srce})) \in \mathcal{U}$ à $(\text{cible}, \text{sol}(\text{cible})) \in \mathcal{U}$ le chemin défini par

$$ca = \{(\text{pb}_{i-1}, \text{sol}(\text{pb}_{i-1})), (r_i, \mathcal{A}_{r_i}), (\text{pb}_i, \text{sol}(\text{pb}_i))\}_{1 \leq i \leq q}$$

Son coût est le coût de cs . Ce coût est nul ssi $srce = cible$ ($q = 0$).

Une adaptation par règles pour l'application jouet.

L'application jouet respecte les hypothèses indiquées plus haut : $\mathcal{U} = \mathcal{U}_{pb} \times \mathcal{U}_{sol}$ avec $\mathcal{U}_{pb} = \mathbb{R}_+^*$ et $\mathcal{U}_{sol} = \mathbb{R}$; un cas source s'écrit $\{(srce, sol(srce))\}$; un cas cible s'écrit $\{cible\} \times \mathcal{U}_{sol}$. Les équations (2) et (3) peuvent se traduire par trois règles d'adaptation :

$$\begin{aligned} - & \begin{cases} srce \underline{r} cible \text{ si } cible = 1/srce \\ sol(cible) = -sol(srce) \end{cases} \\ - & \begin{cases} srce \underline{r} cible \text{ si } cible = 10 \cdot srce \\ sol(cible) = sol(srce) + 1 \end{cases} \\ - & \begin{cases} srce \underline{r} cible \text{ si } cible = 0,1 \cdot srce \\ sol(cible) = sol(srce) - 1 \end{cases} \end{aligned}$$

La première traduit (2), les deux dernières traduisent (3).

Par exemple, si $cible = \frac{0,01}{srce}$, alors en appliquant un chemin de similarité constitué, dans un ordre quelconque, d'une application de la première règle et de deux applications de la troisième règle, on obtient $sol(cible) = -sol(srce) - 2$.

De façon générale, si $cible = 10^p \cdot srce^s$ ($p \in \mathbb{Z}$, $s \in \{-1, +1\}$), $sol(cible) = p + s \cdot sol(srce)$ et le coût d'adaptation sera $p + \frac{1-s}{2}$.

L'avantage de cette approche est qu'elle fournira un résultat précis. Son inconvénient est qu'elle ne permet de résoudre que les problèmes de la forme $10^p \cdot srce^s$ où $p \in \mathbb{Z}$, $s \in \{-1, +1\}$ et $\{(srce, sol(srce))\}$ est un cas source.

5 Lien entre les deux approches de l'adaptation

Les deux approches de l'adaptation présentées brièvement ci-dessus ont été étudiées indépendamment l'une de l'autre. Le but de cette section est l'étude de leurs liens.

5.1 L'adaptation à base de règles est une adaptation par révision

On se place sous les hypothèses de l'adaptation par règles. On peut noter de façon préalable que cette approche de l'adaptation ne tient pas compte des connaissances du domaine, ce qui revient à poser $CD = \mathcal{U}$. Soit d_{RA} , la distance sur $\mathcal{U} = \mathcal{U}_{pb} \times \mathcal{U}_{sol}$ définie par

$$d_{RA}(x, y) = \min\{\text{coût}(ca) \mid ca : \text{chemin d'adaptation de } x \text{ à } y\} \quad (6)$$

où par convention $\min \emptyset = +\infty$. Soit $Source = \{x\}$ un cas source spécifique et $Cible = \{cible\} \times \mathcal{U}_{sol}$ un cas cible avec une partie problème spécifique. Soit $CibleRésolue$ définie par l'équation (4) avec $\dot{+} = \dot{+}_{d_{RA}}$. $y \in CibleRésolue$ ssi il existe un chemin d'adaptation de coût minimal de x à y , ce qui équivaut à dire que $y = (cible, sol(cible))$, avec $sol(cible)$ obtenue par adaptation le long d'un chemin de similarité de coût minimal. Autrement dit, $sol(cible) \in Solutions(cible)$

où $\text{Solutions}(\text{cible})$ est le résultat de l'adaptation par règles de $(\text{srce}, \text{sol}(\text{srce}))$ à cible . Donc $\text{CibleRésolue} = \{\text{cible}\} \times \text{Solutions}(\text{cible})$. Ainsi, la résolution du problème d'adaptation considéré à l'aide des règles d'adaptation coïncide avec la résolution de ce problème d'adaptation à l'aide de l'opérateur de révision $\dagger_{d_{\text{RA}}}$.

Une conséquence de cela est qu'une adaptation par règles, conçue pour des cas sources spécifiques, peut être généralisée immédiatement pour des cas sources généraux : il suffit d'appliquer la $\dagger_{d_{\text{RA}}}$ -adaptation sur de tels cas sources. De la même façon, on peut tenir compte de la présence de connaissances du domaine ($\text{CD} \subsetneq \mathcal{U}$) mais cela suppose que les chemins d'adaptation ca de l'équation (6) soient cohérents avec ces connaissances, autrement dit, pour tout $(\text{pb}_i, \text{sol}(\text{pb}_i))$ de ca , il faut que $(\text{pb}_i, \text{sol}(\text{pb}_i)) \in \text{CD}$.

5.2 Adapter des cas par révision et règles

Les deux approches de l'adaptation présentées ci-dessus présentent des inconvénients.

L'inconvénient principal de l'approche par règles (ou de la $\dagger_{d_{\text{RA}}}$ -adaptation) est qu'elle dépend fortement de la disponibilité des règles d'adaptation. Certains travaux ont pour objectif l'obtention de ces règles, que ce soit par apprentissage automatique (Hanney & Keane (1996)), auprès d'experts (d'Aquin *et al.* (2006)) ou de façon semi-automatique selon les principes de l'extraction des connaissances et de la fouille de données (Craw *et al.* (2006); d'Aquin *et al.* (2007)). Néanmoins rares (ou inexistantes) sont les applications non jouets pour lesquelles les règles d'adaptation disponibles sont suffisantes pour traiter tous les problèmes d'adaptation qui seraient jugés triviaux par un expert. Cette affirmation est difficile à démontrer mais ressort de notre expérience de développement de systèmes de RÀPC.

Formellement, la \dagger -adaptation comprenant l'adaptation par règles, elle ne devrait pas avoir de limitations que n'a pas cette dernière. Néanmoins, cette approche de l'adaptation peut donner à penser qu'il suffit de trouver un opérateur de révision quelconque sur \mathcal{U} et que cela suffit. En fait, la plupart de nos travaux antérieurs sur la \dagger -adaptation étaient fondés sur des opérateurs de révision \dagger_d avec des distances simples (p. ex., la distance de Hamming, la distance de Manhattan sur \mathbb{R}^n ou des distances fondées sur la recherche de plus courts chemins dans des hiérarchies de concepts) et ont donné des résultats satisfaisants, encore qu'aucune évaluation expérimentale systématique n'ait encore été menée. C'est aussi un argument qu'on trouve en théorie de la révision : Datal (1988) décrit un opérateur de révision qu'on peut écrire sous la forme \dagger_{d_H} où d_H est la distance de Hamming non pondérée entre interprétations et justifie ce choix par le fait d'éviter d'introduire un biais (qui consisterait à pondérer un changement sur une variable plus lourdement qu'un changement sur une autre). Ce point de vue de l'adaptation et de la révision est défendable, tant qu'il s'agit de systèmes assez simple (ou de premières versions). Mais pour aller plus loin, ce genre d'opérateurs ne suffit pas toujours. En effet, on peut montrer sous certaines hypothèses techniques (voir la proposition ci-dessous) que si la distance d s'écrit sous la forme d'une somme d'une distance sur \mathcal{U}_{pb} et d'une distance sur \mathcal{U}_{sol} , donc sans un terme qui caractérise le lien problème-solution, alors la \dagger_d -adaptation se ramène à une copie de la solution du cas source, corrigée si nécessaire pour rétablir la cohérence avec les connaissances du domaine. Or ce sera le cas avec les distances de Hamming ou de Manhattan, dès lors que les

attributs sont partitionnés entre attributs problèmes et attributs solution (et le résultat se généralise pour d'autres distances, comme la distance euclidienne).

Proposition 1

En l'absence de connaissances du domaine ($CD = \mathcal{U}$), si on a une séparation problème-solution des cas ($\mathcal{U} = \mathcal{U}_{pb} \times \mathcal{U}_{sol}$, $Cible = \{cible\} \times \mathcal{U}_{sol}$, $Source = P_{srce} \times S_{srce}$), si la distance d sur \mathcal{U} s'écrit comme la somme d'une distance d_{pb} sur \mathcal{U}_{pb} et d'une distance d_{sol} sur \mathcal{U}_{sol} , i.e. si pour tout $x = (x_{pb}, x_{sol}), y = (y_{pb}, y_{sol}) \in \mathcal{U}$,

$$d(x, y) = d_{pb}(x_{pb}, y_{pb}) + d_{sol}(x_{sol}, y_{sol})$$

et enfin si S_{srce} est un fermé de $(\mathcal{U}_{sol}, d_{sol})$ ($\{y_{sol} \in \mathcal{U}_{sol} \mid d_{sol}(S_{srce}, y_{sol}) = 0\} = S_{srce}$) alors la \dagger_d -adaptation de Source pour Cible consiste en une copie de la solution du cas source :

$$CibleRésolue = \{cible\} \times S_{srce}$$

Preuve Soit $y = (y_{pb}, y_{sol}) \in CibleRésolue$. Cela équivaut à $y_{pb} = cible$ et y minimise $d(Source, y)$. Or

$$\begin{aligned} d(Source, y) &= \inf_{x \in Source} d(x, y) \\ &= \inf_{(x_{pb}, x_{sol}) \in P_{srce} \times S_{srce}} [d_{pb}(x_{pb}, cible) + d_{sol}(x_{sol}, y_{sol})] \\ &= \underbrace{\left[\inf_{x_{pb} \in P_{srce}} d_{pb}(x_{pb}, cible) \right]}_{d_{pb}(P_{srce}, cible)} + \underbrace{\left[\inf_{x_{sol} \in S_{srce}} d_{sol}(x_{sol}, y_{sol}) \right]}_{d_{sol}(S_{srce}, y_{sol})} \end{aligned}$$

Donc $y \in CibleRésolue$ si y_{sol} minimise $d_{sol}(S_{srce}, y_{sol})$, ce qui revient à dire que $d(S_{srce}, y_{sol}) = 0$ et donc $y_{sol} \in S_{srce}$. \square

On notera dans la suite d_{CD} une distance s'écrivant comme une somme de d_{pb} et de d_{sol} , comme dans cette proposition. Cette notation se justifie par le fait que seules les connaissances du domaine font que la $\dagger_{d_{CD}}$ -adaptation puisse être autre chose qu'une adaptation par copie de la solution.

Pour résumer ce qui précède, l'adaptation par règles a l'inconvénient de ne pas résoudre suffisamment de problèmes d'adaptation, alors que la $\dagger_{d_{CD}}$ -adaptation (i.e., la \dagger -adaptation avec un opérateur de révision ne gérant pas d'interactions problème-solution) est souvent une simple copie, éventuellement corrigée par CD.

Pour combiner ces deux approches, une première idée est de définir une \dagger_d -adaptation réalisant un choix entre les deux approches :

$$d(x, y) = \min(K_{RA} \cdot d_{RA}(x, y), K_{CD} \cdot d_{CD}(x, y)) \quad \text{pour } x, y \in \mathcal{U}$$

Les coefficients $K_{RA} > 0$ et $K_{CD} > 0$ sont choisis pour doser la priorité entre les deux types d'adaptation. Par exemple, si K_{RA} est petit devant K_{CD} , cela signifie que l'adaptation par règles, quand elle s'applique (i.e., quand $d_{RA}(S_{srce}, cible) < +\infty$), est généralement préférée à la $\dagger_{d_{CD}}$ -adaptation.

Une deuxième idée de combinaison des deux approches de l'adaptation consiste à les composer (dans le même esprit que celui de la combinaison des règles d'adaptation) :

on adapte le cas source par des règles en un cas intermédiaire entre le cas source et le cas cible, puis on applique la $\dagger_{d_{\text{CD}}}$ -adaptation. Cela peut se modéliser par une \dagger_d -adaptation avec

$$d(x, y) = \inf_{z \in \mathcal{U}} K_{\text{RA}} \cdot d_{\text{RA}}(x, z) + K_{\text{CD}} \cdot d_{\text{CD}}(z, y) \quad \text{pour } x, y \in \mathcal{U}$$

(si la borne inférieure de l'expression ci-dessus n'est pas atteinte, on se contentera d'une valeur de z qui donnera une valeur proche de cette borne).

Combinaisons des deux approches pour l'application jouet.

Appliquer la première combinaison consiste à choisir, au cas (d'adaptation) par cas (d'adaptation), une des deux approches. En l'occurrence, l'adaptation par règles, quand elle donne un résultat (i.e., $d_{\text{RA}}(\text{srce}, \text{cible}) < +\infty$, c.-à-d. $\text{cible} = 10^p \cdot \text{srce}^s$ pour un $p \in \mathbb{Z}$ et un $s \in \{-1, +1\}$) est préférable. Cela revient à faire

si $\text{cible} = 10^p \cdot \text{srce}^s$ ($p \in \mathbb{Z}, s \in \{-1, +1\}$)
alors $\text{sol}(\text{cible}) = \min(p + s \cdot \text{sol}(\text{srce}), 0,44(\text{cible} - 1))$
sinon $\text{sol}(\text{cible}) = \min(\text{sol}(\text{srce}), 0,44(\text{cible} - 1))$

Appliquer la deuxième combinaison, sous l'hypothèse $K_{\text{RA}} \ll K_{\text{CD}}$, revient à appliquer les règles d'adaptation en un cas $\{z\}$ le plus proche du cas cible et à appliquer une copie-correction. En l'occurrence, $z = (z_1, z_2)$ où $z_1 = 10^p \cdot \text{srce}^s$ avec p et s choisis de façon à ce que z_1 soit le plus proche possible de cible au sens de d_{CD} . Plus formellement :

$$\begin{aligned} \text{sol}(\text{cible}) &= \min(p + s \cdot \text{sol}(\text{srce}), 0,44(\text{cible} - 1)) \\ \text{où } (p, s) &\text{ minimise } K_{\text{RA}} \left(p + \frac{1-s}{2} \right) + K_{\text{CD}} |\text{cible} - 10^p \cdot \text{srce}^s| \\ &\text{sur } \mathbb{Z} \times \{-1, +1\} \end{aligned}$$

6 Conclusion et perspectives

Cet article compare deux approches de l'adaptation : la \dagger -adaptation et l'adaptation par règles. Il montre comment la seconde peut être incorporée dans la première : les règles d'adaptation modifient la topologie de l'espace des cas et, quand elles ne sont pas applicables pour un problème d'adaptation donné, l'adaptation s'appuie uniquement sur les connaissances du domaine.

Ce travail en est à une phase encore préliminaire. La première direction de recherches sera d'étudier, sous certaines hypothèses, un algorithme réalisant cette adaptation combinant connaissances du domaine et règles d'adaptation.

Du point de vue de l'ingénierie des connaissances, il devrait être intéressant d'étudier comment faire évoluer la distance d d'une \dagger_d -adaptation, pour qu'elle incorpore de nouvelles règles d'adaptation, acquises ou apprises lors d'échecs du raisonnement (vis-à-vis de l'utilisateur), à l'image de l'acquisition opportuniste de connaissances d'adaptation (Cordier (2008)). Partant d'une distance vérifiant l'hypothèse de la proposition 1,

on s'attend à ce que d , au cours du temps, incorpore de plus en plus de liens entre les composantes de problèmes et les composantes de solutions.

Enfin, ce travail s'inscrit dans la volonté de formaliser l'adaptation et d'unifier les approches de l'adaptation. Il doit se poursuivre par la comparaison de la \dagger -adaptation avec d'autres approches de l'adaptation.

Remerciements

Les auteurs tiennent à remercier les relecteurs pour leurs intéressantes remarques. Nous espérons avoir su les exploiter pour améliorer la qualité de cet article.

Références

- ALCHOURRÓN C. E., GÄRDENFORS P. & MAKINSON D. (1985). On the Logic of Theory Change : partial meet functions for contraction and revision. *Journal of Symbolic Logic*, **50**, 510–530.
- COJAN J. & LIEBER J. (2008). Conservative Adaptation in Metric Spaces. In *Proceedings of the 9th Conference on Case-Based Reasoning (ECCBR-08)*, Lecture Notes in Artificial Intelligence 5239, p. 135–149 : Springer.
- COJAN J. & LIEBER J. (2009). Belief Merging-based Case Combination. In *Case-Based Reasoning Research and Development (ICCBR 2009)*, p. 105–119.
- COJAN J. & LIEBER J. (2010). An Algorithm for Adapting Cases Represented in an Expressive Description Logic. In SPRINGER, Ed., *Proceedings of ICCBR-2010 (to be published)*.
- CORDIER A. (2008). *Interactive and Opportunistic Knowledge Acquisition in Case-Based Reasoning*. PhD thesis, Université Lyon 1, France.
- COULON D., BOIVIEUX J.-F., BOURRELLY L., BRUNEAU L., CHOURAQUI E., DAVID J.-M., LU C. R., PY M., SAVELLI J., SÉROUSSI B. & VRAIN C. (1990). Le raisonnement par analogie en intelligence artificielle. In B. BOUCHON-MEUNIER, Ed., *Actes des 3^{èmes} journées nationales du PRC-GDR Intelligence Artificielle*, p. 45–88.
- CRAW S., WIRATUNGA N. & ROWE R. C. (2006). Learning adaptation knowledge to improve case-based reasoning. *Artificial Intelligence*, **170**(16-17), 1175–1192.
- DALAL M. (1988). Investigations into a theory of knowledge base revision : Preliminary report. In *AAAI*, p. 475–479.
- D'AQUIN M., BADRA F., LAFROGNE S., LIEBER J., NAPOLI A. & SZATHMARY L. (2007). Case Base Mining for Adaptation Knowledge Acquisition. In M. M. VELOSO, Ed., *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07)*, p. 750–755 : Morgan Kaufmann, Inc.
- D'AQUIN M., LIEBER J. & NAPOLI A. (2006). Adaptation Knowledge Acquisition : a Case Study for Case-Based Decision Support in Oncology. *Computational Intelligence (an International Journal)*, **22**(3/4), 161–176.
- HANNEY K. & KEANE M. T. (1996). Learning Adaptation Rules From a Case-Base. In I. SMITH & B. FALTINGS, Eds., *Advances in Case-Based Reasoning – Third European Workshop, EWCBR'96*, LNAI 1168, p. 179–192 : Springer Verlag, Berlin.

- KATSUNO H. & MENDELZON A. (1991). Propositional knowledge base revision and minimal change. *Artificial Intelligence*, **52**(3), 263–294.
- LIEBER J. (2002). Strong, Fuzzy and Smooth Hierarchical Classification for Case-Based Problem Solving. In F. VAN HARMELEN, Ed., *Proceedings of the 15th European Conference on Artificial Intelligence (ECAI-02)*, Lyon, France, p. 81–85 : IOS Press, Amsterdam.
- LIEBER J. (2007). Application of the Revision Theory to Adaptation in Case-Based Reasoning : the Conservative Adaptation. In *Proceedings of the 7th International Conference on Case-Based Reasoning (ICCBR-07)*, Lecture Notes in Artificial Intelligence 4626, p. 239–253. Belfast : Springer.
- MELIS E., LIEBER J. & NAPOLI A. (1998). Reformulation in Case-Based Reasoning. In B. SMYTH & P. CUNNINGHAM, Eds., *Fourth European Workshop on Case-Based Reasoning, EWCBR-98*, Lecture Notes in Artificial Intelligence 1488, p. 172–183 : Springer.
- RIESBECK C. K. & SCHANK R. C. (1989). *Inside Case-Based Reasoning*. Hillsdale, New Jersey : Lawrence Erlbaum Associates, Inc.
- ROUGEGREZ S. (1994). Similarity evaluation between observed behaviours for the prediction of processes. In S. WESS, K.-D. ALTHOFF & M. M. RICHTER, Eds., *Topics in Case-Based Reasoning – First European Workshop (EWCBR'93)*, Kaiserslautern, Lecture Notes in Artificial Intelligence 837, p. 155–166 : Springer Verlag, Berlin.
- VELOSO M. M. (1994). *Planning and Learning by Analogical Reasoning*. LNAI 886. Springer Verlag, Berlin.