Apprentissage de réseaux de préférences *ceteris paribus*⋆

Frédéric Koriche[1] et Bruno Zanuttini[2]

[1] LIRMM, CNRS UMR 5526, Université Montpellier II
frederic.koriche@lirmm.fr

[2] GREYC, CNRS UMR 6072, Université de Caen Basse-Normandie, ENSICAEN
bruno.zanuttini@info.unicaen.fr

Résumé : Nous étudions l'acquisition de réseaux de préférences *ceteris paribus* (CP-nets), selon le paradigme de l'apprentissage exact avec requêtes défini par Angluin. Dans ce contexte, l'apprenant cherche à découvrir un CP-net cible représentant le modèle cognitif de l'utilisateur en lui posant des questions. Nous montrons que la classe générale des CP-nets booléens n'est pas apprenable avec requêtes d'équivalence et d'appartenance. Puis nous donnons un algorithme montrant que l'importante sous-classe des CP-nets *acycliques complets de taille polynomiale en le nombre de variables* est apprenable avec requêtes d'équivalence et d'appartenance. Ce résultat encourageant indique que les réseaux de « petit degré » utilisés en pratique sont efficacement identifiables par le biais de l'apprentissage interactif.

1 Introduction

La notion de *préférence* joue un rôle central en intelligence artificielle pour la conception d'agents autonomes capables d'assister les humains dans la prise de décision. Par exemple, dans le commerce électronique, l'agent peut aider l'utilisateur à choisir un produit en triant la base de données selon ses préférences. En planification routière, l'agent peut aussi aider l'utilisateur en lui présentant des itinéraires qui optimisent ses préférences en matière de consommation d'énergie, de durée du trajet, de trafic routier, etc. Comme l'ont souligné plusieurs auteurs (Keeney & Raiffa, 1976; French, 1986; Ha & Haddawy, 1998; Kahneman & Tversky, 2000), un des problèmes fondamentaux dans ce type d'applications est d'*acquérir* (ou *éliciter*) les préférences d'un utilisateur dans une représentation permettant la fois de décrire fidèlement le modèle cognitif de l'utilisateur et de raisonner efficacement sur ce modèle.

Parmi les nombreux formalismes de représentation des préférences proposés dans la littérature, les réseaux de préférences *ceteris paribus* ou *CP-nets* ont fait l'objet d'un

⋆Ce travail a été réalisé dans le cadre du projet CANAR, financé par l'ANR (ANR-06-BLAN-0383-02).

intérêt croissant, en proposant une représentation graphique à la fois intuitive et efficace (Boutilier *et al.*, 1999, 2001, 2004; Brafman & Domshlak, 2002; Goldsmith *et al.*, 2005; Wilson, 2004, 2006). Intuitivement, un CP-net peut être vu comme un réseau de croyances où les tables de probabilités conditionnelles associées à chaque variable sont simplement remplacées par des tables de préférences conditionnelles de la forme « toutes choses étant égales par ailleurs, je préfère le littéral p à sa négation $\overline{p}$ dans les états du monde vérifiant le motif t ». D'un point de vue algorithmique, certaines classes de CP-nets, telles que celle des CP-nets booléens arborescents, peuvent supporter en temps polynomial divers processus de raisonnement comme le test de dominance qui consiste à savoir, parmi deux états du monde x et x', si x' est préféré ou non à x.

Dans cet article, nous étudions l'acquisition de CP-nets booléens selon le paradigme de l'apprentissage exact avec requêtes introduit par Angluin (1988). Ce paradigme a suscité beaucoup d'intérêt dans la littérature, en posant un cadre formel au processus d'apprentissage interactif et en permettant d'acquérir efficacement, par interaction, des classes expressives utilisées en représentation des connaissances telles que les formules existentielles conjonctives (Haussler, 1989), les théories de Horn (Angluin *et al.*, 1992), ou encore certaines logiques de description (Frazier & Pitt, 1996).

Dans le cadre des CP-nets, le modèle cible n'est pas utilisé pour classifier des données mais plutôt pour les comparer entre elles. Un *exemple* est alors une paire (x, x') d'états du monde. L'apprenant cherche à identifier le modèle de l'utilisateur en lui posant des questions. Dans une *requête d'équivalence*, l'apprenant présente un CP-net à l'utilisateur et lui demande s'il correspond à son modèle. Si le CP-net n'est pas correct, l'apprenant reçoit un contre-exemple (x, x') qui lui permet de réviser son hypothèse. Dans une *requête d'appartenance*, l'apprenant présente un exemple (x, x') à l'utilisateur et lui demande si effectivement x est préféré à x'. Le but est d'identifier le modèle cible en utilisant le moins de ressources possibles (requêtes et temps de calcul).

Cet article établit des résultats d'apprenabilité pour deux classes de réseaux de préférences. Nous commençons par montrer que la classe générale des CP-nets booléens n'est pas apprenable avec requêtes d'équivalence et d'appartenance. Pour cette classe, qui autorise un nombre arbitraire de parents par sommet du graphe, la dimension de Vapnik-Chervonenkis (Blumer *et al.*, 1989) s'avère en effet au moins exponentielle en le nombre de variables. Par contraste, nous démontrons par un algorithme que l'importante sous-classe des CP-nets acycliques complets de taille polynomiale en le nombre de variables est efficacement apprenable avec requêtes d'équivalence et requêtes d'appartenance.

Comme la taille d'un CP-net augmente de manière exponentielle avec le degré de son graphe, la plupart des modèles exploités en pratique sont fondés sur des graphes de petit degré (Boutilier *et al.*, 2004). Ainsi, notre résultat positif indique que les réseaux de préférences avec « petit degré » sont effectivement identifiables par le biais de l'apprentissage interactif. Notamment, les réseaux arborescents nécessitent seulement un nombre quadratique de requêtes pour être identifiés.

2 Réseaux de préférences ceteris paribus

Si v est une variable booléenne, les *littéraux* sur v sont le littéral *positif* v et le littéral *négatif* $\overline{v}$. Si p est un littéral, alors $\overline{p}$ désigne le littéral opposé. Par exemple, si p est le littéral $\overline{v}$, alors $\overline{p}$ est le littéral v. Un *terme* t est un ensemble de littéraux, vu comme leur conjonction. L'ensemble des variables apparaissant dans t est noté $var(t)$. Si V est un ensemble de variables, un terme t est dit *maximal* sur V s'il contient exactement un littéral par variable de V, c'est à dire $var(t) = V$.

Étant donné un ensemble de variables V, un *état* x du monde est un terme maximal sur V. L'espace des états est noté X. Si t est un terme sur V, on dit que x *satisfait* t si l'on a $t \subseteq x$. Enfin, si x est un état et t un terme, on note $x[t]$ l'état obtenu à partir de x en remplaçant les littéraux de x définis sur $var(t)$ par les littéraux de t. Formellement, $x[t] = \{p \mid p \in t \text{ ou } p \in x \text{ et } \overline{p} \notin t\}$. Par exemple, si x est l'état $\{v_1, v_2, \overline{v}_3, \overline{v}_4\}$ et t est le terme $v_1 \wedge \overline{v}_2 \wedge v_3$, alors $x[t]$ est l'état $\{v_1, \overline{v}_2, v_3, \overline{v}_4\}$.

Nous présentons maintenant les CP-nets booléens. Soient v une variable booléenne et t un terme avec $v \notin var(t)$. Une *règle de préférence ceteris paribus* (CP-règle) sur v sachant t est une expression de la forme $t : v > \overline{v}$ ou $t : \overline{v} > v$. Intuitivement, $t : v > \overline{v}$ signifie que, toutes choses étant égales par ailleurs, v est préférable à $\overline{v}$ dans les états du monde satisfaisant t.

Soient v une variable booléenne et Pa un ensemble fini de variables booléennes avec $v \notin Pa$. Une *table de préférences ceteris paribus* (CP-table) pour v selon Pa, notée $cpt(v)$, est la donnée d'au plus une CP-règle sur v sachant t, pour chaque terme maximal t sur Pa. La CP-table $cpt(v)$ est dite *complète* si elle associe exactement une CP-règle sur v sachant t à chaque terme maximal t sur Pa.

Définition 1 (CP-net)
Soit $V = \{v_1, \ldots, v_n\}$ un ensemble de variables booléennes. Un réseau de préférences ceteris paribus *(CP-net) sur V est la donnée d'un graphe orienté G sur V et, pour tout $i = 1, \ldots, n$, d'une table $cpt(v_i)$ selon l'ensemble Pa_i des parents de v_i dans G.*

Un CP-net N sur le graphe G est dit *complet* si chacune de ses tables est complète. Il est dit *acyclique* si G est acyclique, et *arborescent* si G est une forêt, c'est-à-dire si chaque variable a au plus un arc entrant/parent dans G, et G ne contient pas de cycle.

Le *degré* de N, noté $deg(N)$, est défini par le degré du graphe de N, c'est-à-dire le nombre maximum de parents d'une variable de N. La *taille* d'un CP-net N, notée $||N||$, est définie comme le nombre de CP-règles distinctes apparaissant dans N. Si l'on note $V = \{v_1, \ldots, v_n\}$ l'ensemble des variables de N, on a donc $deg(N) = \max_{v_i \in V} |Pa_i|$, et $||N|| \leq \sum_{i=1}^{n} 2^{|Pa_i|}$, avec une égalité stricte si N est complet.

Exemple 2 (tenue de soirée)
Le CP-net est décrit dans la partie gauche de la figure 1. Il fait intervenir les trois variables v, p et c, pour le choix de la veste, celui du pantalon et celui de la chemise. De manière inconditionnelle, notre utilisatrice préfère le noir ($\overline{v}$) au blanc (v) pour la couleur de la veste, et de même pour celle du pantalon. Si le pantalon et la veste sont de la même couleur, elle préfère le rose $\overline{c}$ au blanc c pour la couleur de la chemise. En revanche, si le pantalon et la chemise sont de couleurs différentes, elle choisira une

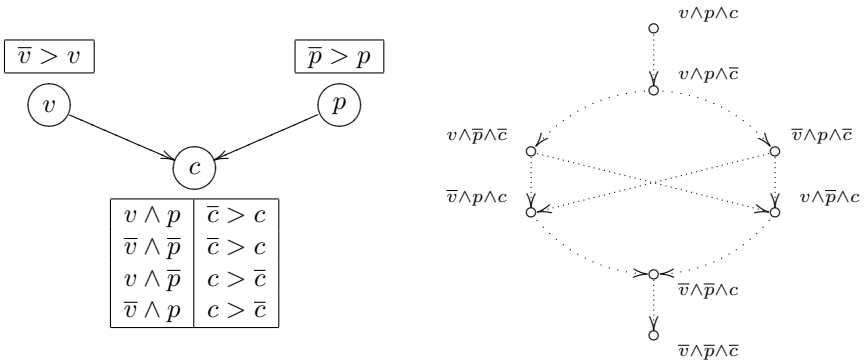

FIG. 1 – Scénario « tenue de soirée » avec à gauche un CP-net, et à droite un préordre partiel sur l'espace d'états qui satisfait le CP-net.

chemise blanche. Notons que le CP-net est acyclique, booléen, complet, et de degré 2. *La partie droite de la figure 1 donne un préordre partiel satisfaisant ce CP-net.*

D'un point de vue sémantique, un CP-net N représente un ensemble de fonctions d'utilité sur les états du système. Rappelons qu'une fonction d'utilité f associe à chaque état x de X un réel $f(x)$ appelé utilité de x. Nous disons que f *satisfait* une CP-règle $t : v > \overline{v}$ si pour tout état x dans X satisfaisant t, nous avons $f(x[v]) > f(x[\overline{v}])$. La satisfaction d'une CP-règle $t : \overline{v} > v$ est définie de manière duale. Nous disons que f *satisfait* une CP-table $cpt(v)$ sur une variable booléenne v si f satisfait chaque règle de $cpt(v)$. Enfin, nous disons que f *satisfait* un CP-net N sur un ensemble de variables booléennes $V = \{v_1, \ldots, v_n\}$ si f satisfait chaque table $cpt(v_i)$ de N.

Définition 3 (dominance)
Soient V un ensemble de variables, N un CP-net sur V, et x, x' deux états sur X. Alors on dit que x domine *x' selon N, et on note $x \succ_N x'$, si $f(x) > f(x')$ pour toute fonction d'utilité f sur X satisfaisant N.*

Nous dirons que deux CP-nets $\hat{N}, N$ sont *équivalents* si pour tout couple d'états (x, x'), x domine x' selon $\hat{N}$ si et seulement s'il le domine selon N.

La proposition suivante caractérise la dominance entre deux états ne diffèrant qu'en un littéral. La preuve est omise par manque de place, mais elle découle de la caractérisation de la dominance par les séquences améliorantes (Boutilier *et al.* (2004)).

Proposition 4
Soit N un CP-net acyclique, x un état, et p un littéral. Alors $x[p] \succ_N x[\overline{p}]$ si et seulement s'il existe dans N une règle de la forme $t : p > \overline{p}$ où $t \subseteq x$.

3 Apprentissage exact de CP-nets

L'apprentissage exact introduit par Angluin (1988) peut être vu comme un jeu de découverte entre l'utilisateur et l'apprenant. Le premier dispose d'un modèle qui est inconnu du second. Il peut néanmoins lui apporter une aide limitée, en répondant à certaines requêtes. Le but du jeu, pour l'apprenant, est de trouver le modèle de l'utilisateur en dépensant le moins de ressources (requêtes et temps de calcul) possible.

Nous appelons *exemple* une paire d'états (x, x') qui ne diffèrent que sur la valeur d'une seule variable. D'un point de vue cognitif, cette restriction est justifiée par le fait que de tels exemples sont relativement simples à trouver pour un utilisateur humain ; ils correspondent à des situations du type « pour cette voiture, je la préfèrerais en rouge plutôt qu'en blanc », où la couleur est une des multiples variables décrivant la voiture. D'un point de vue algorithmique, cette restriction est aussi justifiée par le fait qu'en appliquant la proposition 4, le test de dominance entre x et x' est linéaire pour tout réseau acyclique N, alors qu'il deviendrait NP-difficile si x et x' étaient deux états arbitraires (Boutilier *et al.*, 2004). Enfin, notons que dans tout CP-net complet N, si x et x' diffèrent en une seule variable, on a toujours $x \succ_N x'$ ou $x' \succ_N x$, et que deux CP-nets sont équivalents si et seulement s'ils le sont sur de tels couples d'états.

Une *classe* de CP-nets est un ensemble $\mathcal{N}$ de CP-nets. Nous disons qu'un exemple (x, x') *appartient* à un réseau cible N dans $\mathcal{N}$ si x domine x' dans N. En se basant sur cette notation, nous pouvons donc voir tout CP-net comme un concept. Etant donné un ensemble E d'exemples, soit $\mathcal{N}(E)$ l'ensemble de tous les CP-nets dans $\mathcal{N}$ identifiant une partie de E, c'est à dire $\mathcal{N}(E) = \{E' \subseteq E : \exists N \in \mathcal{N} : E' = E \cap N\}$. La *dimension de Vapnik-Chervonenkis*, ou *VC-dimension*, d'une classe $\mathcal{N}$ de CP-nets, notée $dim(\mathcal{N})$, est le cardinal du plus grand ensemble d'exemples pour lequel nous avons $|\mathcal{N}(E)| = 2^{|E|}$.

Les deux types de requêtes généralement employées en apprentissage exact sont les requêtes d'*appartenance*, utilisées pour connaître la valeur d'un exemple, et les requêtes d'*équivalence*, utilisées pour verifier si l'hypothèse apprise correspond au modèle cible.

Plus précisément, pour une requête d'appartenance $\mathrm{MQ}(x, x')$, l'apprenant fournit un couple d'états (x, x') à l'utilisateur, qui répond oui si (x, x') appartient à N, c'est à dire $x \succ_N x'$, et non sinon. Pour une requête d'équivalence $\mathrm{EQ}(\hat{N}, N)$, l'apprenant fournit un CP-net $\hat{N}$ à l'utilisateur, qui répond oui si N et $\hat{N}$ sont équivalents, et non sinon. Dans ce dernier cas, l'utilisateur fournit de plus un *contre-exemple* (x, x') pour lequel $\hat{N}$ et N impliquent une préférence différente.

Définition 5 (apprentissage exact)
Un algorithme A est dit d'apprentissage avec requêtes d'équivalence et d'appartenance *pour une classe $\mathcal{N}$ de CP-nets, s'il existe un polynôme p tel que pour tout CP-net N de $\mathcal{N}$ sur n variables, après $p(n)$ requêtes d'équivalence et d'appartenance, A converge vers un CP-net $\hat{N}$ de $\mathcal{N}$ qui est équivalent à N. A est dit de plus* efficace *s'il existe un polynôme q tel que la complexité temporelle de A est bornée par $q(n)$. Dans ce cas, $\mathcal{N}$ est dite* apprenable avec requêtes d'équivalence et d'appartenance.

Notons que nous définissons l'efficacité de l'apprentissage en fonction du nombre de variables, et non de la taille du réseau cible.

4 Résultats d'apprenabilité

Après une présentation du formalisme des CP-nets et du modèle d'apprentissage, nous pouvons passer au cœur de l'article en examinant trois résultats d'apprenabilité. Nous commençons par montrer que la classe générale $\mathcal{N}_{\mathrm{BIN}}$ des CP-nets booléens complets n'est pas apprenable avec un nombre de requêtes d'équivalence et d'appartenance polynomial en le nombre de variables. Notons que ce résultat n'est pas surprenant, étant donné que l'on cherche a apprendre un objet de taille exponentielle. Mais le résultat est vrai même sans limite sur le temps de calcul (seulement sur le nombre de requêtes).

Lemme 6
La VC-dimension de $\mathcal{N}_{\mathrm{BIN}}$ est au moins exponentielle en le nombre de variables.

Preuve À chaque ensemble de variables $\{v_1, \ldots, v_n\}$, nous faisons correspondre un ensemble E_n contenant 2^{n-1} exemples tels que pour toute partie de E_n, il existe dans $\mathcal{N}_{\mathrm{BIN}}$ un CP-net de taille n contenant exactement cette partie. Par définition de la VC-dimension, on en déduit donc $dim(\mathcal{N}_{\mathrm{BIN}}) \geq 2^{n-1}$. Soit E_n l'ensemble de tous les exemples de la forme $(\{p_1, \ldots, p_{n-1}, v_n\}, \{p_1, \ldots, p_{n-1}, \overline{v}_n\})$ tels que pour chaque $i \in \{1, \ldots, n-1\}$, p_i est un littéral sur la variable v_i. Donc E_n est l'ensemble des paires d'états du monde qui ne diffèrent qu'en v_n. Cet ensemble est de taille 2^{n-1}, mais toute partie E'_n de E_n peut être uniquement identifiée par un CP-net (acyclique et complet) contenant des préférences inconditionnelles quelconques sur $v_1, \ldots, v_{n-1}$, et pour v_n, la CP-règle $p_1 \wedge \cdots \wedge p_{n-1} : v_n > \overline{v}_n$ si $(\{p_1, \cdots, p_{n-1}, v_n\}, \{p_1, \cdots, p_{n-1}, \overline{v}_n\})$ est dans E'_n, et la règle opposée sinon. □

Théorème 7 (non-apprenabilité des CP-nets de degré quelconque)
La classe $\mathcal{N}_{\mathrm{BIN}}$ n'est pas apprenable avec requêtes d'équivalence et d'appartenance. Ceci s'applique aussi pour la classe $\mathcal{N}_{\mathrm{ACY}}$ des CP-nets booléens complets acycliques.

Preuve (Maass & Turán, 1992, Théorème 6.5) montrent que la VC-dimension d'une classe de concepts constitue une borne inférieure au nombre minimum de requêtes d'équivalence et d'appartenance nécessaires à l'identification exacte des concepts de la classe. Comme la VC-dimension de $\mathcal{N}_{\mathrm{BIN}}$ est au moins exponentielle en le nombre de variables, il est donc impossible d'apprendre cette classe avec requêtes d'équivalence et d'appartenance. Enfin, comme la preuve de la VC-dimension s'applique aussi pour les réseaux acycliques, la classe $\mathcal{N}_{\mathrm{ACY}}$ n'est donc pas non plus apprenable avec requêtes d'équivalence et d'appartenance. □

À présent, nous montrons que la classe $\mathcal{N}^{poly}_{\mathrm{ACY}}$ des CP-nets (booléens) acycliques complets de taille polynomiale en le nombre de variables est identifiable efficacement en utilisant des requêtes d'appartenance et d'équivalence. Dans ce cadre, l'apprenant sait que le réseau est complet et acyclique, mais *ne connaît pas* son degré. Ceci démarque notre résultat de l'approche directe pour l'élicitation de petits réseaux, approche consistant à considérer tous les ensembles de parents possibles ou à utiliser la connaissance du graphe des parents.

L'algorithme, présenté dans la figure 2, démarre en affectant une préférence inconditionnelle sur les valeurs de chaque variable, en se basant sur la préférence sur cette

Algorithme 2 : Apprentissage de CP-nets complets acycliques de taille polynomiale

début
  Initialisation des CP-tables
  $\hat{N} \leftarrow \emptyset$
  $x_0 \leftarrow$ l'état mettant tous les littéraux à faux
  pour chaque *variable* v *dans* V **faire**
    $Pa(v) \leftarrow \emptyset$
    si $\mathrm{MQ}(x_0[v], x_0[\overline{v}])$ **alors** $cpt(v) \leftarrow \{\emptyset : v > \overline{v}\}$
    sinon $cpt(v) \leftarrow \{\emptyset : \overline{v} > v\}$
    $\hat{N} \leftarrow \hat{N} \cup \{cpt(v)\}$
  Raffinement des CP-tables jusqu'à identification
  tant que $(x[p], x[\overline{p}]) \leftarrow \mathrm{EQ}(\hat{N}, N) = non$ **faire**
    On note v *la variable de* p
    Soit $t : \overline{p} > p$ la règle de $\hat{N}$ telle que $t \subseteq x$
    $pa \leftarrow$ EXTRAIREPARENT$(p, t : \overline{p} > p, x)$
    Mise à jour des parents de v *dans* $\hat{N}$
    $\hat{N} \leftarrow \hat{N} \setminus \{cpt(v)\}$
    $Pa(v) \leftarrow Pa(v) \cup \{pa\}$
    Mise à jour de la CP-table de v *dans* $\hat{N}$
    $cpt(v) \leftarrow \emptyset$
    pour chaque *terme maximal* t *sur* $Pa(v)$ **faire**
      si $\mathrm{MQ}(x_0[t \wedge v], x_0[t \wedge \overline{v}])$ **alors** $cpt(v) \leftarrow cpt(v) \cup \{t : v > \overline{v}\}$
      sinon $cpt(v) \leftarrow cpt(v) \cup \{t : \overline{v} > v\}$
    $\hat{N} \leftarrow \hat{N} \cup \{cpt(v)\}$
  retourner $\hat{N}$
fin

variable, *tout littéral étant à faux par ailleurs* (état x_0) ; cet état x_0 servira de référence si la préférence inconditionnelle est invalidée par un contre-exemple ultérieur[1].

Par la suite, lorsqu'il reçoit un contre-exemple de la forme $(x[p], x[\overline{p}])$, l'algorithme l'utilise pour déterminer un nouveau parent de la variable v sur laquelle p est formé, en s'appuyant sur l'état x et la règle qui a été mise en échec par le contre-exemple. Puis il complète la CP-table sur v avec le nouvel ensemble de parents, en utilisant des requêtes d'appartenance (toujours avec l'état de référence x_0).

La procédure EXTRAIREPARENT est specifiée dans la figure 3. L'idée est de tester, avec des requêtes d'appartenance, si chaque nouvelle variable n'apparaissant pas dans la condition de la règle mise en échec est un parent de la variable apparaissant dans la tête de la règle. Le lemme suivant détaille et prouve la correction de cet algorithme.

[1]Tout autre état, y compris un état choisi par l'utilisateur, pourrait ainsi servir de référence.

Algorithme 3 : Découverte d'un nouveau parent d'une variable

Entrées : littéral p formé sur v, règle mise en échec $t : \overline{p} > p$, état x
Sorties : parent pa de v tel que $pa \notin var(t)$
début
  $x_c \leftarrow x$;
  pour chaque *variable* $pa \notin var(t)$ *telle que* $pa \neq v$ *et* $x \models pa$ **faire**
    si $\mathrm{MQ}(x[\overline{pa} \wedge p], x[\overline{pa} \wedge \overline{p}])$ **alors** $x_c \leftarrow x_c[\overline{pa}]$;
    sinon retourner pa
fin

Lemme 8
Soient N un CP-net booléen acyclique complet sur un ensemble de variables V, p un littéral sur $v \in V$, t un terme sur $V \setminus \{v\}$, et x un état tel que $x \models t \wedge p$ et $x[p] \succ_N x[\overline{p}]$. Supposons que l'état x_0 qui met tous les littéraux de V à faux est tel que $x_0[t \wedge \overline{p}] \succ_N x_0[t \wedge p]$. Alors l'algorithme 3 retourne un parent pa de v dans N, avec $pa \notin var(t)$.

Preuve Remarquons tout d'abord que l'algorithme maintient l'invariant $x_c[p] \succ_N x_c[\overline{p}]$. Donc l'algorithme ne peut pas terminer sans retourner une variable, car cela signifierait qu'il a transformé, littéral par littéral, l'état x en l'état $x_0[t]$. Du fait de l'invariant, on aurait donc $x_0[t \wedge p] \succ_N x_0[t \wedge \overline{p}]$, ce qui contredit l'hypothèse.

Soit donc pa la variable retournée. On a $x_c[pa \wedge p] \succ_N x_c[pa \wedge \overline{p}]$ par l'invariant et $x_c[\overline{pa} \wedge \overline{p}] \succ_N x_c[\overline{pa} \wedge p]$ par construction, donc pa est un parent de v. □

Lemme 9
Pour toute variable v, l'algorithme 2 maintient un réseau $\hat{N}$ tel que l'ensemble des parents de v dans $\hat{N}$ est inclus dans celui du réseau cible N. En particulier, puisque N est acyclique, $\hat{N}$ l'est également.

Preuve Cet invariant est démontré par induction sur chaque raffinement de l'hypothèse $\hat{N}$. La propriété est clairement vérifiée après l'initialisation des ensembles de parents à $\emptyset$. Donc, supposons par hypothèse d'induction que la propriété soit vraie pour $\hat{N}$ juste avant de recevoir un nouveau contre-exemple. Puisque la requête d'équivalence a fourni un nouveau contre-exemple $(x[p], x[\overline{p}])$, et que le réseau $\hat{N}$ est complet par construction, cela signifie qu'il implique $x[\overline{p}] \succ_{\hat{N}} x[p]$. D'autre part, comme $\hat{N}$ est acyclique par hypothèse d'induction, il contient bien une règle de la forme $t : \overline{p} > p$ telle que $t \subseteq x$ (proposition 4). Puisque cette règle a été construite sur x_0, on conclut en utilisant le lemme 8 que l'ensemble Pa retourné par la procédure d'extraction est inclus dans les parents de v dans N. □

Théorème 10 (apprenabilité des CP-nets acycliques polynomiaux)
L'algorithme 2 identifie exactement les CP-nets acycliques booléens complets de taille polynomiale en le nombre n de variables. Il utilise pour cela $O(n \log n)$ requêtes d'équivalence et $O(n^2 \log n)$ requêtes d'appartenance, et sa complexité temporelle est en $O(n \log n(c_{\mathrm{EQ}} + nc_{\mathrm{MQ}}))$, où c_{EQ} est la complexité d'une requête d'équivalence et c_{MQ} celle d'une requête d'appartenance.

Preuve Si l'algorithme s'arrête, par définition d'une requête d'équivalence, alors il a bien identifié le réseau cible. Pour montrer qu'il s'arrête, on utilise le lemme 9 et le fait qu'à chaque mise à jour d'un ensemble de parents, cet ensemble croît. L'union des ensembles de parents étant finie, l'algorithme converge donc.

Concernant l'analyse de complexité, rappelons que la taille du réseau cible, booléen et complet, est en $O(2^d)$ où d est le degré du réseau. De plus, puisque sa taille est polynomiale en n il s'ensuit que d est en $O(\log n)$. Le nombre d'appels EQ est en $O(dn)$, puisqu'à chaque fois on découvre un nouveau parent d'une variable. Pour chaque contre-exemple reçu, l'algorithme commence à extraire un nouveau parent avec $O(n)$ appels MQ et utilise ensuite une requête MQ pour chaque ligne de la nouvelle table, donc $O(n+2^d)$ requêtes d'appartenance. Puisque d est en $O(\log n)$, la complexité temporelle est bien en $O(n \log n(c_{\mathrm{EQ}} + nc_{\mathrm{MQ}}))$. □

Corollaire 11 (apprenabilité des CP-nets arborescents)
L'algorithme 2 identifie exactement les CP-nets arborescents (booléens complets) en utilisant seulement $O(n)$ requêtes d'équivalence et $O(n^2)$ requêtes d'appartenance.

5 Discussion

Cette étude se situe à l'intersection de deux domaines de l'intelligence artificielle : le raisonnement sur les préférences et l'apprentissage exact. Nous avons démontré que, d'un côté, la classe générale des CP-nets booléens complets, même acycliques, n'est pas apprenable avec requêtes d'équivalence et d'appartenance, et de l'autre, la sous-classe importante des CP-nets acycliques complets de taille polynomiale est apprenable avec seulement des requêtes d'équivalence et d'appartenance.

Concernant les travaux relatifs à ce cadre d'étude, l'apprentissage, ou élicitation, de préférences a suscité un intérêt récent dans la littérature. Notamment, Domshlak & Joachims (2005) emploient des machines à vecteurs de support pour apprendre des fonctions booléennes à seuil permettant de discriminer des préférences. Dans un contexte plus proche, Blum *et al.* (2004) utilisent l'apprentissage exact avec requêtes pour éliciter des préférences numériques pour les problèmes d'enchères. Dans un cadre symbolique, Sachdev (2007) étudie l'apprentissage de préférences dans un cadre proche du nôtre, mais plus général. Les pistes qu'il évoque pour les CP-nets sont donc des approches tirant moins parti de leur structure. Enfin, Athienitou & Dimopoulos (2007) étudient l'apprentissage de CP-nets minimaux (au sens des ensembles de parents) à partir d'exemples, en utilisant une énumération par niveaux des ensembles de parents.

Nous pensons que cette étude préliminaire sur l'apprenabilité des CP-nets ouvre la voie à plusieurs perspectives intéressantes. Notre algorithme 2 peut être optimisé pour éviter de recalculer les CP-tables ; une borne inférieure sur le nombre de requêtes nous permettrait aussi de confirmer l'optimalité. D'autre part, l'utilisation des requêtes d'équivalence peut être sujette à question dans les situations où l'utilisateur ne peut pas toujours déterminer si son modèle correspond au réseau fourni par l'apprenant. Une alternative, suggérée par Angluin (1988), est de simuler les appels EQ par des requêtes stochastiques. Une autre approche est d'explorer si certaines classes de CP-nets ne né-

cessitent, en fait, que des requêtes d'appartenance. Enfin, le problème de savoir si des réseaux non triviaux sont PAC apprenables, sans utiliser des requêtes, reste à explorer.

Références

ANGLUIN D. (1988). Queries and concept learning. *Machine Learning*, **2**(4), 319–342.

ANGLUIN D., FRAZIER M. & PITT L. (1992). Learning conjunctions of Horn clauses. *Machine Learning*, **9**, 147–164.

ATHIENITOU F. & DIMOPOULOS Y. (2007). Learning CP-Networks : A preliminary investigation. In *3rd Multidisciplinary Workshop on Advances in Preference Handling (M-PREF'07)*.

BLUM A., JACKSON J. C., SANDHOLM T. & ZINKEVICH M. (2004). Preference elicitation and query learning. *Journal of Machine Learning Research*, **5**, 649–667.

BLUMER A., EHRENFEUCHT A., HAUSSLER D. & WARMUTH M. K. (1989). Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, **36**(4), 929–965.

BOUTILIER C., BACCHUS F. & BRAFMAN R. I. (2001). UCP-networks : A directed graphical representation of conditional utilities. In *17th Conference in Uncertainty in Artificial Intelligence (UAI)*, p. 56–64 : Morgan Kaufmann.

BOUTILIER C., BRAFMAN R. I., DOMSHLAK C., HOOS H. H. & POOLE D. (2004). CP-nets : A tool for representing and reasoning with conditional *ceteris paribus* preference statements. *Journal of Artificial Intelligence Research*, **21**, 135–191.

BOUTILIER C., BRAFMAN R. I., HOOS H. H. & POOLE D. (1999). Reasoning with conditional ceteris paribus preference statements. In *15th Conference on Uncertainty in Artificial Intelligence (UAI)*, p. 71–80 : Morgan Kauffmann.

BRAFMAN R. I. & DOMSHLAK C. (2002). Introducing variable importance tradeoffs into CP-nets. In *18th Conference in Uncertainty in Artificial Intelligence (UAI)*, p. 69–76 : Morgan Kaufmann.

DOMSHLAK C. & JOACHIMS T. (2005). Unstructuring user preferences : Efficient non-parametric utility revelation. In *21st Conference in Uncertainty in Artificial Intelligence (UAI)*, p. 169–177 : AUAI Press.

FRAZIER M. & PITT L. (1996). Classic learning. *Machine Learning*, **25**(2-3), 151–193.

FRENCH S. (1986). *Decision Theory*. Halsted Press.

GOLDSMITH J., LANG J., TRUSZCZYNSKI M. & WILSON N. (2005). The computational complexity of dominance and consistency in CP-nets. In *19th International Joint Conference on Artificial Intelligence (IJCAI)*, p. 144–149 : Professional Book Center.

HA V. A. & HADDAWY P. (1998). Toward case-based preference elicitation : Similarity measures on preference structures. In *14th Conference on Uncertainty in Artificial Intelligence (UAI)*, p. 193–201 : Morgan Kaufmann.

HAUSSLER D. (1989). Learning conjunctive concepts in structural domains. *Machine Learning*, **4**, 7–40.

KAHNEMAN D. & TVERSKY A. (2000). *Choices, Values, and Frames*. Cambridge University Press.

KEENEY R. L. & RAIFFA H. (1976). *Decisions with Multiple Objectives : Preferences and Value Tradeoffs*. Wiley.

MAASS W. & TURÁN G. (1992). Lower bounds methods and spearation results for online-learning models. *Machine Learning*, **9**, 107–145.

SACHDEV M. (2007). On Learning of *Ceteris Paribus* Preference Theories. Master's thesis, Graduate Faculty of North Carolina State University.

WILSON N. (2004). Extending CP-nets with stronger conditional preference statements. In *19th National Conference on Artificial Intelligence (AAAI)*, p. 735–741 : AAAI Press.

WILSON N. (2006). An efficient upper approximation for conditional preference. In *Proc. of the 17th European Conference on Artificial Intelligence (ECAI)*, p. 472–476 : IOS Press.