

Nouvelle taxonomie des méthodes de classification basée sur l'Analyse de Concepts Formels

Marwa Trabelsi¹ — Nida Meddouri^{1,2} — Mondher Maddouri³

¹ Laboratoire d'Informatique, Programmation, Algorithmique et Heuristique
Université d'El Manar, Tunisie
trabelsimarou@live.com

² Département Technologies Multimédia et Web
Institut Supérieur des Langues Appliquées et Informatique de Beja
Université de Jendouba, Tunisie
nida.meddouri@gmail.com

³ Department of Computer Sciences, Community College of Hinakya
Taibah University, Medina
Kingdom of Saudi Arabia
mondher.maddouri@fst.rnu.tn

RÉSUMÉ. Des diverses approches ont été proposées dans la classification supervisée de données parmi lesquelles l'approche basée sur l'Analyse de Concepts Formels. Cet article présente une vue d'ensemble des méthodes de classification basée sur l'Analyse de Concepts Formels. Nous traitons ce sujet en proposant une nouvelle taxonomie de ces méthodes et en présentant une étude comparative basée sur la complexité théorique de ces dernières.

ABSTRACT. Various approaches have been proposed in supervised classification, among them the approach based on Formal Concept Analysis. This paper presents an overview of classification methods based on Formal Concepts Analysis. We address this issue by proposing a new taxonomy of these methods and presenting a comparative study based on the theoretical complexity.

MOTS-CLÉS : Fouille de Données, Classification Supervisée, Analyse de Concepts Formels

KEYWORDS : Data Mining, Supervised Classification, Formal Concept Analysis

1. Introduction

Le volume de données connaît une évolution considérable et perpétuelle. Plusieurs travaux se sont focalisés sur l'extraction de connaissances à partir des données. Fayyad et al [21] ont défini l'extraction de connaissances à partir des données comme étant l'acquisition de nouvelles connaissances qui sont potentiellement utiles à partir des faits cachés au sein de grandes quantités de données. L'un des processus fondamentaux de l'extraction de connaissances repose sur la fouille de données, en particulier la classification supervisée. Ce processus peut être réalisé par les réseaux de neurones, les arbres de décisions, les réseaux bayésiens, les machines à vecteur de support ou encore l'Analyse de Concepts Formels [7]. Le choix de l'Analyse de Concepts Formels repose sur la capacité de cette dernière de traiter de grandes quantités de données et de simplifier la prédiction des classes [17, 7]. La classification basée sur l'Analyse de Concepts Formels consiste à construire des modèles appelés classifieurs à partir des données permettant de prédire des classes pour les futures données. Elle vise à découvrir tous les regroupements possibles de concepts et d'extraire ensuite les règles de classification selon les concepts générés à partir des données [17]. L'opération est réalisée en deux phases : **une phase d'apprentissage** dans laquelle un classifieur est construit pour décrire un ensemble prédéterminé de classes d'objets à partir d'un ensemble d'apprentissage et **une phase de classification** où le classifieur construit est utilisé pour associer une classe à chaque nouvel objet.

Cet article est organisé comme suit : la section 2 présente les notions de base de l'Analyse de Concepts Formels. Nous abordons, ensuite, la nouvelle taxonomie de méthodes de classification supervisée. Enfin, dans la section 3 nous proposons une étude comparative des méthodes présentées.

2. Classification supervisée basée sur l'Analyse de Concepts Formels

2.1. Analyse de Concepts Formels

L'Analyse de Concepts Formels est développée autour d'une communauté de mathématiciens. Ensuite, elle a attiré progressivement les chercheurs en informatique et a servi de fondement théorique à de nombreuses applications. Des multiples méthodes d'analyse de données et de représentation de connaissances ont été proposées. Ces méthodes traitent l'information sous la forme d'une hiérarchie de concepts formels [2].

Un contexte formel relie un ensemble fini d'objets G à un ensemble fini d'attributs M grâce à une relation binaire I . Il est représenté sous forme d'un triplet $K=(G, M, I)$ où I vaut 1 quand un objet g vérifie l'attribut m avec g et m appartenant respectivement à G et M notée $I(g, m)=1$ [2]. Le contexte formel peut ainsi être illustré par un tableau de deux dimensions où on présente les objets sur les lignes et les attributs sur les colonnes. La case (i, j) indique la valeur de la relation entre l'objet g_i et l'attribut m_j avec i varie de 1 à n et j varie de 1 à m (n est le nombre des objets et m est le nombre des attributs). À partir d'un contexte formel $K=(G, M, I)$, nous pouvons extraire tous les concepts formels possibles. L'ensemble ordonné¹ de tous les concepts peut être organisé sous forme d'un

1. Un treillis est un ensemble ordonné dans lequel toute paire d'éléments admet une borne supérieure et une borne inférieure

treillis appelé treillis complet de concepts formels, dit treillis de Galois [2]. Les méthodes de classification basées sur l'Analyse de Concepts Formels adoptent généralement une approche exhaustive ou une approche combinatoire. Dans ce qui suit, nous détaillons ces approches en donnant un survol des méthodes existantes de chaque approche.

2.2. Les méthodes de classification exhaustive

Les méthodes exhaustives se servent d'un seul classifieur et d'un regroupement de concepts formels durant les phases de la classification supervisée de données². Cependant, elles varient entre elles selon la taille du treillis (complet ou demi) utilisé.

2.2.1. Méthodes avec treillis complet

GRAND [5, 18], RULEARNER [10, 18], GALOIS [4, 18], CBALATTICE [1], NAVIGALA [11], HMCS-FCA-SC [8] et SPFC [6] ont utilisé des treillis complets comme espace de recherche. Ces méthodes valident les caractéristiques associées aux concepts niveau par niveau dans les treillis. La navigation dans le treillis commence à partir du concept minimal où tous les concepts sont considérés comme des candidats.

GRAND³ et GALOIS sont les premières méthodes qui utilisent des treillis complets. GRAND, lors de la phase d'apprentissage, organise l'information extraite d'un contexte formel sous forme d'un treillis complet à l'aide d'un algorithme incrémental qui considère le contexte ligne par ligne (colonne par colonne) et construit les treillis par ajout successif de concepts. Il réalise la mise à jour des treillis par l'ajout de nouveaux noeuds et par suppression des connexions redondantes [19]. Ensuite, elle applique les règles les plus spécifiques [5] à chaque objet. GALOIS construit, à l'image de GRAND, un treillis complet de façon incrémentale et ascendante. Dans la phase de classification, le système calcule la similarité entre le nouvel objet et chaque concept qui correspond au nombre de propriétés du concept vérifiées par l'objet [4].

D'autres méthodes comme NAVIGALA et RULEARNER sont par la suite issues de GRAND. NAVIGALA⁴ a comme particularité lors de la construction du treillis, l'utilisation d'un contexte d'objets décrits par des vecteurs numériques de taille fixe. Ces vecteurs sont stockés dans une table discrète qui devient par la suite binaire [11].

De même, RULEARNER utilise un treillis complet pour la recherche des règles de classification. Elle construit le treillis de la même façon que GRAND. Durant la classification, elle se sert du vote majoritaire pour la détermination des classes des objets [10].

CBALATTICE construit un treillis complet de concepts et applique des règles d'association dans le but d'extraire des règles de classification. La méthode est incrémentale et progressive, toute augmentation du nombre d'objets, d'attributs et des classes peut être manipulée de manière très efficace [1].

HMCS-FCA-SC⁵ a également eu recours à la construction d'un treillis complet afin de créer un modèle de classification hiérarchique. Durant la classification, elle emploie une mesure de similarité cosinus⁶ entre le nouvel exemple et les concepts sélectionnés pour la classification des données [8].

2. La plupart des méthodes citées ont utilisé des échantillons de la base d'UCI (<http://archive.ics.uci.edu/ml/>)

3. Graph-based induction

4. Navigation into Galois Lattice

5. Hierarchical Multi-label Classifier System - Formal Concept Analysis with Similarity Cosine

6. Consiste à calculer la similarité entre deux vecteurs à n dimensions en déterminant le cosinus de l'angle entre eux.

Après la construction du treillis, SPFC⁷ affecte à chaque concept un score qui indique si les concepts sont convenables pour la génération des règles de classification. SPFC cherche, ensuite, les voisins des concepts pertinents (ayant les scores les plus élevés). Les objets inconnus seront classés dans les classes auxquelles appartiennent leurs voisins [6].

Les limites des méthodes exhaustives résident dans la construction d'un classifieur ayant une capacité de prédiction insuffisante et une complexité exponentielle de leurs algorithmes d'apprentissage en terme de temps et de ressources mémoires utilisés.

2.2.2. Méthodes avec demi treillis

Pour remédier à ce problème, d'autres travaux comme LEGAL [9, 18], CIBLE [18], CLNN & CLNB [23, 18], IPR [12], CLANN [14] et CITREC [3] ont eu recours aux demi treillis. Un demi treillis de concepts est une structure mathématique permettant de représenter une partie du treillis de concepts de façon sélective [3].

LEGAL⁸ construit un sup-demi treillis⁹ de concepts en s'appuyant sur certains paramètres d'apprentissage. Durant la phase d'apprentissage, elle construit un ensemble ordonné de concepts en se basant sur la classe de chaque instance. Les instances positives et négatives sont les instances étiquetées par une classe positive ou une classe négative dans le contexte formel. Au cours de la classification LEGAL applique le vote majoritaire [9].

CIBLE¹⁰ est réalisée en deux étapes successives : elle commence par la construction d'un sup-demi treillis à partir d'un contexte binaire puis il se sert d'une mesure de similarité pour la classification des nouvelles instances [18].

CLNN & CLNB¹¹ construisent un sup-demi treillis de façon descendante. Ensuite, elles incorporent respectivement un classifieur Bayésien naïf et un classifieur plus proche voisin dans chaque noeud du demi treillis construit. CLNN & CLNB utilisent la même technique de vote qui est le vote majoritaire lors de la phase de classification [23].

CITREC¹² construit le treillis à partir d'un contexte réduit contenant seulement une instance représentative de chaque classe [3]. Dans la phase de classification, CITREC utilise aussi le vote majoritaire comme les méthodes CLNN & CLNB.

CLANN¹³ construit un sup-demi treillis durant l'apprentissage en traitant les données qui possèdent seulement deux classes. Puis, elle utilise ce demi treillis pour construire un réseau de neurone qui réalise la classification [14].

IPR¹⁴ introduit la notion de couverture. Elle fait recours à un algorithme glouton pour construire la couverture de concepts. L'algorithme choisit les concepts pertinents et chaque concept est déterminé grâce à une optimisation locale de la fonction d'entropie [12]. Pour chaque nouvel objet, IPR cherche les règles dont leur prémisse coïncide avec les attributs et applique la règle la plus pondérée pour cet objet.

La classification basée sur un demi treillis se déroule de la même manière que celle basée sur un treillis complet. Les méthodes basées sur les demi treillis permettent, par contre, de minimiser l'ensemble de règles de classification générées en gardant les plus pertinentes ce qui engendre un gain considérable aux niveaux du temps et d'apprentissage mais une perte d'information en même temps.

7. Classification by Selecting Plausible Formal Concepts in a Concept Lattice

8. Learning with Galois Lattice

9. Un sup demi-treillis est un ensemble ordonné dans lequel deux éléments quelconques x et y admettent toujours une borne supérieure

10. Concept Induction Based Learning

11. Concept Lattices Nearest Neighbors and Concept Lattices Naive Bayes

12. Classification Indexée par le treillis de concepts

13. Concept Lattice-based Artificial Neural Network

14. Induction of Production Rules

Concernant les méthodes exhaustives citées auparavant, nous observons plusieurs inconvénients. D'une part, leurs complexités sont élevées. D'autre part, elles utilisent un classifieur faible et unique. Ainsi, on observe l'absence de l'aspect adaptatif dans la classification. Par conséquent, les chercheurs s'orientent vers les méthodes combinatoires.

2.3. Les méthodes de classification combinatoire

Dans le but d'améliorer la performance d'un classifieur unique (estimé faible) qui est adopté par les méthodes exhaustives, les méthodes combinatoires génèrent un ensemble de classifieurs et les combinent par les techniques de votes.

Plusieurs méthodes ont été proposées dans ce cadre : des méthodes qui réalisent l'apprentissage séquentiel telles que BFC [15], BNC [16] et bien d'autres basées sur l'apprentissage parallèle comme DNC [17], FCA-BRG [13] et RMCS [22].

L'apprentissage séquentiel consiste à générer des classifieurs d'une manière séquentielle c'est-à-dire, un classifieur, n'est généré qu'après la génération de son prédécesseur. À titre d'exemple, BFC¹⁵ construit à partir d'un contexte formel une couverture formée seulement des concepts pertinents. Cette dernière se base sur le principe du boosting qui est une approche adaptative basée sur l'utilisation de plusieurs classifieurs du même modèle [20]. L'idée de BFC consiste à affecter des poids égaux aux exemples d'apprentissage parmi lesquels un sous-ensemble est sélectionné à l'aide d'un tirage probabiliste. À ce stade, un concept pertinent est extrait à partir du sous ensemble en sélectionnant l'attribut qui minimise la fonction d'entropie de Shannon¹⁶. BFC permet ensuite de générer une règle de classification déduite à partir du concept pertinent (extraite du sous ensemble) et de faire une mise à jour des poids aux exemples d'apprentissage. Cette procédure est appliquée récursivement pour construire finalement le classifieur final [15].

BNC¹⁷ procède de la même manière que BFC dans la génération des classifieurs et le traitement de données d'apprentissage. En revanche, contrairement à BFC qui effectue le traitement de données binaires, BNC manipule des données nominales dans le but d'éviter la perte d'information issue de la représentation binaire [16].

L'apprentissage parallèle basé sur le principe de Dagging [20], consiste à diviser l'ensemble de données en plusieurs groupes à partir desquels les classifieurs sont produits. DNC¹⁸ traite des données nominales et se déroule comme suit : un tirage de données est effectué afin de créer des groupes disjoints contenant des données stratifiés. Un classifieur de concept nominal [16] est ensuite construit dans chaque groupe. Enfin, la méthode utilise la technique de vote pour définir une combinaison de sortie des classifieurs [16].

Dans l'apprentissage parallèle, FCA-BRG¹⁹ commence également par la division de la base initiale en des sous ensembles de données. Ces derniers sont, ensuite, utilisés pour la génération des sous contextes formels afin d'extraire les règles de classification. Un algorithme génétique sera enfin appliqué pour sélectionner les meilleures règles [13].

RMCS²⁰ construit des classifieurs basés sur les voisins. Ils ne réalisent la classification correcte d'un objet s'ils ont classé correctement ses voisins. RMCS commence par la construction d'une table de classification à partir d'un contexte formel (les objets du contexte utilisé sont privés d'un ensemble d'objets test). Dans cette table, RMCS affecte

15. Boosting Formal Concepts

16. L'entropie de Shannon, est une fonction mathématique qui correspond à la quantité d'information contenue ou délivrée par une source d'information.

17. Boosting Nominal Concepts

18. Dagging Nominal Concepts

19. FCA Based Rule Generator

20. Recommender based Multiple Classifier System

les classifieurs aux objets existants dans le contexte. Ensuite, RMCS cherche les voisins des objets de test à l'aide d'une métrique de similarité, puis elle sélectionne des classifieurs qui ont le nombre maximal de voisins trouvés. Les classifieurs sélectionnés sont ainsi recommandés pour la classification [22].

3. Discussion

Comme nous l'avons indiqué, les méthodes de classification basées sur l'Analyse de Concepts Formels sont regroupées en deux catégories principales : exhaustives et d'autres combinatoires. Les méthodes de chaque catégorie se distinguent entre elles sur certains aspects mais en partageant d'autres. Les méthodes exhaustives ont comme point commun la génération d'un seul classifieur ordinaire pour la classification des objets.

Le tableau 1 présente les méthodes exhaustives évoquées précédemment. Afin de dégager les particularités de chaque méthode, nous avons eu recours à cinq critères qui nous semblent les plus déterminants. Le critère le plus important de cette comparaison consiste à calculer un ordre de complexité en fonction des paramètres de classification utilisés. D'autres critères ont été utilisés tels que la structure de concepts, le type de données, la méthode de sélection de concepts utilisée lors de la classification et la méthode de classification qui désigne la manière d'affectation des nouveaux objets aux classes.

ystème	structure	données	sélection	classification	complexité
GRAND	treillis complet	binaires	cohérence maximale	vote majoritaire	$O(2^k * k^4)$ $k = \min(m, n)$
CIBLE	demi treillis	numériques	fonction de sélection	K-PPV	$O(L * m^3)$ $ L = \text{demi treillis}$
IPR	couverture	binaires	entropie	règles pondérées	$O(n^2 * m^2 * nm)$
CITREC	demi treillis	binaires	support	vote	$O(2^m * n)$
CLANN	demi treillis	binaires	algorithmes heuristiques	réseau de neurone	$O(2^{\min(m, n)})$
HMCS-FCA-SC	treillis complet	nominales	meilleurs concepts	mesure de similarité	$O(2^m + n2^m)$

Tableau 1. Comparaison théorique des méthodes exhaustives

Comme le montre le tableau 1, les méthodes exhaustives possèdent une complexité exponentielle. Cela est dû principalement à la navigation dans la totalité de l'espace de recherche contrairement aux méthodes combinatoires qui distribuent le processus de classification sur des multiples classifieurs. Le problème est ainsi décomposé en plusieurs sous-problèmes. Le tableau 2 distingue les méthodes combinatoires. Pour des raisons comparatives nous avons utilisé les cinq critères du tableau 1 en y ajoutant la méthode de combinaison de classifieurs employés. Les tableaux 1 et 2 montrent que GRAND, IPR, CITREC, CLANN, BFC et RMCS opèrent à partir des données binaires, tandis que BNC et DNC traitent des données nominales. En revanche, CIBLE se distingue par rapport aux précédentes de sa capacité de traiter des données numériques. BNC et DNC utilisent le gain informationnel dans la sélection des concepts, tandis que IPR et BFC se servent de l'entropie de Shannon. Quant à CLANN, elle utilise des algorithmes heuristiques pour la sélection.

ystème	structure	données	sélection	combinaison	classification	complexité
BFC	couverture	binaires	entropie	boosting	vote pondéré	$O(n\log(n)+nm)$
BNC	couverture	nominales	gain informationnel	boosting	vote pondéré	$O(n\log(n)+nm)$ m =attribut nominal
DNC	couverture de concepts pertinents	nominales	gain informationnel	dagging	Vote majoritaire	$O(n^2)$ n = sous échantillon stratifié [17]
RMCS	treillis complet	binaires	distance euclidienne	dagging	maximum de voisin	$O(nm\log(n))$

Tableau 2. Comparaison théorique des méthodes combinatoires

Concernant la phase de classification, GRAND, CITREC et DNC utilisent le vote majoritaire. Le vote pondéré a été appliqué par IPR, BFC et BNC. En revanche, CLANN diffère des autres méthodes par l'utilisation du réseau de neurone.

La technique de combinaison (cf. section 2.3) a joué un rôle important dans l'optimisation de la complexité²¹. En effet, les méthodes combinatoires qui génèrent des classifieurs de manière séquentielle ont une complexité polynomiale logarithmique. De même, les méthodes qui génèrent des classifieurs parallèles arrivent à une complexité comparable de l'ordre de $nm\log(n)$ pour la méthode RMCS et de n pour DNC.

4. Conclusion

Dans cet article, nous nous sommes intéressés par la classification supervisée de données basée sur l'Analyse de Concepts Formels. Nous avons présenté dans un premier temps les méthodes de classification exhaustive qui se divisent en des méthodes basées sur des treillis complets et des méthodes basées sur des demi treillis. Dans un deuxième temps, nous avons décrit les méthodes de classification combinatoire qui elles-mêmes se décomposent en des méthodes basées sur l'apprentissage séquentiel et des méthodes basées sur l'apprentissage parallèle. Nos perspectives reposent sur la complexité et s'orientent vers les méthodes combinatoires qui offrent une complexité plus raisonnable, en particulier les méthodes qui génèrent des classifieurs parallèles.

5. Bibliographie

- [1] A. GUPTA, N. KUMAR, V. BHATNAGAR, « Incremental classification rules based on association rules using formal concept analysis », *Machine Learning and Data Mining in Pattern Recognition*, vol. 10, n° 11-20, 2005.
- [2] B. GANTER, G. STUMME, R. WILLE, « Formal concept analysis : foundations and applications », *Springer Science Business Media*, vol. 3626, 2005.
- [3] B. DOUAR, C. LATIRI, Y. SLIMANI, « Approche hybride de classification supervisée à base de treillis de Galois : application à la reconnaissance de visages », *Extraction et Gestion des Connaissances*, n° 309-320, 2008.
- [4] C. CARPINETO, G. ROMANO, « Concept data analysis : Theory and applications », *Livre, John Wiley Sons*, vol. 23, 2004.

21. Notez que 'n' est le nombre d'objets et 'm' le nombre d'attributs

- [5] G. OOSTHUIZEN, « The use of a lattice in knowledge processing », *Thesis, University of Strathclyde*, 1988.
- [6] I. MADORI, Y. AKIHITO, « Classification by Selecting Plausible Formal Concepts in a Concept Lattice », *Formal Concepts Analysis meets information Retrieval*, vol. 14, n° 22-35, 2013.
- [7] J. POELMANS, D. IGNATOV, G. DEDENE, « Formal concept analysis in knowledge processing : A survey on applications », *Expert systems with applications*, vol. 40(16), n° 6538-6560, 2013.
- [8] M. FERRANDIN, J. NIEVOLA, F. ENEMBRECK, E. SCALABRIN, K. KREDENS, B. AVILA, « Hierarchical Classification Using FCA and the Cosine Similarity Function », *International Conference on Artificial Intelligence*, vol. 6, n° 281-287, 2013.
- [9] M. LIQUIERE, E. MEPHU NGUIFO, « Legal : learning with galois lattice », *Journées Françaises sur l'Apprentissage*, n° 93-113, 1990.
- [10] M. SAHAMI, « Learning classification rules using lattices » *European Conference on Machine Learning*, n° 343-346, 1995.
- [11] M. VISANI, K. BERTET, J. OGIER, « Navigala : An original symbol classifier based on navigation through a galois lattice », *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 25, n° 449-473, 2011.
- [12] M. MADDOURI, « Towards a machine learning approach based on incremental concept formation », *Intelligent Data Analysis*, vol. 8, n° 267-280, 2004.
- [13] M. CINTRA, M. MONARD, H. CAMARGO, « FCA-BASED RULE GENERATOR, a framework for the genetic generation of fuzzy classification systems using formal concept analysis. », *In Fuzzy Systems (FUZZ-IEEE)*, n° 1-8, 2015.
- [14] N. TSOPZÉ, E. MEPHU NGUIFO, G. TINDO, « CLANN : Concept Lattice-based Artificial Neural Network for Supervised Classification » *Concept Lattice and their applications*, vol. 331, 2007.
- [15] N. MEDDOURI, M. MADDOURI, « Boosting formal concepts to discover classification rules », *Next-Generation Applied Intelligence*, n° 501-510, 2009.
- [16] N. MEDDOURI, M. MADDOURI, « Adaptive learning of nominal concepts for supervised classification », *Knowledge-Based and Intelligent Information and Engineering Systems*, n° 121-130, 2010.
- [17] N. MEDDOURI, H. KHOUFI, M. MADDOURI, « Parallel Learning and Classification for Rules based on Formal Concepts », *Knowledge-Based and Intelligent Information and Engineering Systems*, vol. 35, n° 358-367, 2014.
- [18] P. NJIWOUA, E. MEPHU NGUIFO, « Treillis de Concepts et Classification Super-visée. », *Technique et Science Informatiques*, vol. 24(4), n° 449-488, , 2005.
- [19] R. GODIN, R. MISSAOUI, H. ALAOU, « Incremental concept formation algorithms based on Galois (concept) lattices », *Appeared in Computational Intelligence*, vol. 11(2), n° 246-267, 1995.
- [20] S. KOTSIANTI, D. KANELLOPOULOS, « Combining bagging, boosting and dagging for classification problems », *Knowledge-Based Intelligent Information and Engineering Systems*, vol. 7, n° 493-500, 2007.
- [21] U. FAYYAD, G. PIATETSKY-SHAPIRO, P. SMITH, « Advances in knowledge discovery and data mining », *National Conference on Artificial Intelligence*, vol. 2, n° 2, 1996.
- [22] Y. KASHNITSKY, D. IGNATOV, « Can FCA-based Recommender System Suggest a Proper Classifier ? », *What can FCA do for Artificial Intelligence*, vol. 2, n° 2, 2015.
- [23] Z. XIE, W. HSU, Z. LIU, M. LEE, « Concept lattice based composite classifiers for high predictability », *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 14(2-3), n° 143-156, 2002.