Analyse critique des algorithmes EDA dans le cadre de l'apprentissage de structure de réseaux Bayésiens

Grégory Thibault, Alexandre Aussem, Stéphane Bonnevay

Analyse critique des algorithmes EDA dans le cadre de l'apprentissage de structure de réseaux Bayésiens

Grégory Thibault* — **Alexandre Aussem*** — **Stéphane Bonnevay****

* *Université de Lyon, laboratoire LIESP*
** *Université de Lyon, laboratoire LIRIS*

Bâtiment le Nautibus,
8, Bd Niels Bohr
69622 Villeurbanne Cedex, France

gthibaul@bat710.univ-lyon1.fr
aaussem@univ-lyon1.fr
bonnevay@univ-lyon1.fr

RÉSUMÉ. *Récemment, une nouvelle classe d'algorithmes d'optimisation baptisée EDA (pour* estimation of distribution algorithms*) a montré son efficacité sur le problème de l'apprentissage de structure de réseaux Bayésiens. Pour cela, les chercheurs ont comparé les EDA avec des algorithmes génétiques. Dans cet article nous proposons de comparer les EDA avec un algorithme à base de score reconnu comme performant : GES. L'objectif de ce travail est de réaliser une analyse critique des EDA dans ce cadre.*

ABSTRACT. *Recently, a new class of optimization algorithms called EDA (stand for* estimation of distribution algorithms*) has shown its effectiveness on the problem of learning Bayesian networks structures. For this, the researchers compared the EDA with genetic algorithms. We propose here to compare the EDA with an algorithm-based scoring recognized as powerful: GES. The aim of this work is to achieve a critical analysis of EDA in this context.*

MOTS-CLÉS : *réseaux Bayésiens, apprentissage de structure, EDA, GES*

KEYWORDS: *Bayesian networks, structure learning, EDA, GES*

1. Introduction

La recherche sur les réseaux Bayésiens a connu un grand essor ces dernières années, en particulier sur l'apprentissage automatique de la structure à partir de données (Buntine, 1996). Ce problème très documenté est un problème *NP-difficile* (Chickering, 1996).

Les **EDA** (ou ***E**stimation of **D**istribution **A**lgorithms*) (Blanco *et al.*, 2003) sont des algorithmes qui ont fait leurs preuves dans de nombreux domaines (Larrañaga *et al.*, 2001), y compris sur l'apprentissage de réseaux Bayésiens. Ainsi, Blanco *et al.* (2003), Romero *et al.* (2004) et Thibault *et al.* (2007) présentent plusieurs applications de ces algorithmes et montrent empiriquement leur efficacité en les comparant avec des approches génétiques (Larrañaga *et al.*, 1996). Dans une précédente étude (Thibault *et al.*, 2007), nous avons réalisé une optimisation du paramétrage de l'algorithme afin d'en augmenter la performance.

Nous nous proposons ici d'évaluer la qualité des résultats de cette approche en comparant avec l'algorithme **GES** (ou ***G**reedy **E**quivalent **S**earch*) (Chickering, 2002b), un algorithme très performant utilisant l'espace des équivalents de *Markov*.

2. Apprentissage à base de score

Heckerman *et al.* (1997) discutent de l'intérêt des méthodes à base de score. Elles permettent notamment l'incorporation aisée de connaissances *a priori* (Cheng *et al.*, 2002), un travail plus efficace avec les données manquantes, et la possibilité de combiner plusieurs modèles pondérés (*model averaging*) (Friedman *et al.*, 2000) rend l'étape d'inférence bien moins dépendante des erreurs éventuellement commises pendant l'apprentissage de la structure et permet de mixer des modèles provenant de plusieurs sources.

2.1. *Scores*

Les scores **AIC** (***A**kaike **I**nformation **C**riterion*) (Akaike, 1970) et **BIC** (***B**ayesian **I**nformation **C**riterion*) (Schwarz, 1978), deux scores dérivés du **PML** (***P**enalized **M**aximum **L**ikelihood*) sont composés de deux critères inconciliables : le maximum de vraisemblance (par rapport aux données) et la complexité du modèle. Les structures maximisant ces deux critères sont respectivement le graphe complet (au sens orienté), et le graphe vide.

$$PML(\mathcal{B} \mid \mathcal{D}) = \log \mathcal{L}(\mathcal{D} \mid \theta^{MV}, \mathcal{B}) - f(N) \cdot Dim(\mathcal{B}) \qquad [1]$$

Où $f(N) = 1$ pour **AIC**, $\frac{log(N)}{2}$ pour **BIC**. Dans le cas de données discrètes finies[1] :

$$\log \mathcal{L}(\mathcal{D} \mid \theta^{MV}, \mathcal{B}) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \cdot \log(\frac{N_{ijk}}{N_{ij}}) \quad [2]$$

$$Dim(\mathcal{B}) = \sum_{i=1}^{n} (r_i - 1) \cdot q_i \quad [3]$$

Les scores dits « *Bayésiens* », par exemple **BD** (***B**ayesian **D**irichlet*) (Cooper *et al.*, 1992), estiment la probabilité d'une structure *a posteriori*. Dans le cas des données statiques, on peut simplifier le calcul en utilisant la formule de Bayes :

$$p(\mathcal{B}|\mathcal{D}) = \frac{p(\mathcal{B}, \mathcal{D})}{p(\mathcal{D})} \propto p(\mathcal{B}, \mathcal{D}) = \int_{\theta} \mathcal{L}(\mathcal{D} \mid \theta, \mathcal{B}) \cdot p(\theta \mid \mathcal{B}) \cdot d\theta \quad [4]$$

Une dérivation est possible lorsque les paramètres suivent une distribution de *Dirichlet* et que les données sont **IID** (indépendants et uniformément répartis) :

$$BD(\mathcal{B} \mid \mathcal{D}) = p(\mathcal{B}) \cdot \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(N_{ij} + \alpha_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(N_{ijk} + \alpha_{ijk})}{\Gamma(\alpha_{ijk})} \quad [5]$$

Plus tard ce critère est adapté aux équivalents de *Markov* pour obtenir **BDe** (Heckerman *et al.*, 1995), puis avec des *a priori* uniformes, ce qui donne **BDeu** (Buntine, 1991).

Il existe d'autres scores, comme **MDL** (***M**inimum **D**escription **L**ength*) (Bouckaert, 1993) dont le principe est de minimiser la taille de codage nécessaire à la description du modèle et des données relativement au modèle. Notons enfin l'utilisation de scores

1. notations :

$\mathcal{B}$	le modèle à scorer,
$\mathcal{D}$	la base d'exemples d'apprentissage,
θ^{MV}	les paramètres obtenus par *maximum de vraisemblance*,
$\mathcal{L}$	la mesure de vraisemblance (ici *Kullback-Leibler*),
$f(n)$	le coefficient de pénalisation,
Dim	la dimension du modèle,
n	le nombre de variables,
q_i	le nombre de valuations possibles des parents de X_i,
r_i	le nombre de modalités de la variable X_i,
N_{ijk}	le nb de cas où X_i prend sa k^{e} valeur et ses parents leur j^{e} ,
α_{ijk}	l'*a priori* sur le paramètre θ_{ijk},
$p(\mathcal{B})$	l'*a priori* sur la structure $\mathcal{B}$.

adaptés aux recherches par arcs, *i.e.*, qui utilisent des poids sur les arcs (au lieu des nœuds), comme l'information mutuelle (Chow *et al.*, 1968).

D'une façon générale, le score définit l'objectif de la recherche : les critères **PML** sont utilisés lorsque le modèle recherché doit être efficace en inférence tout en minimisant sa complexité, les scores Bayésiens (**BD**, **BDe** *etc.*) favorisent les structures offrant le meilleur *a posteriori*, *etc*.

En pratique, les scores sont utilisés lorsqu'ils réunissent certaines propriétés.

– La DÉCOMPOSABILITÉ — *i.e.*, la possibilité de calculer un score local en chaque nœud — garantit des temps de calcul moindres lors de modifications locales.

– La propriété d'ÉQUIVALENCE garantit que deux structures d'une même classe d'équivalence de *Markov* auront les mêmes scores. Cette propriété est particulièrement utile lorsque la recherche utilise l'espace des équivalents de *Markov*.

– La CONSISTANCE fait qu'un modèle contenant la distribution — contenue dans les données — obtiendra un meilleur score qu'un modèle ne la contenant pas. De plus cette propriété discrimine les modèles contenants tous deux la distribution mais dont le nombre de paramètres est différent. Il existe une variante locale de cette propriété : la consistance locale qui permet à **GES** d'améliorer localement les structures.

2.2. *Méthodes*

Certains des algorithmes de recherche utilisent des heuristiques de limitation de l'espace de recherche, comme le font les algorithmes **K2** (Cooper *et al.*, 1992) ou **MWST** (***M**aximum **W**eight **S**panning **T**ree*) (Chow *et al.*, 1968). D'autres effectuent leur recherche directement dans l'espace complet à l'aide de métaheuristiques. L'algorithme glouton, ou ***G**reedy **S**earch* (Chickering *et al.*, 1995), progresse à l'aide d'opérations locales, mais converge dans des maxima locaux. Des techniques plus évoluées ont alors été utilisées :

– *iterated hill-climbing* (Campos *et al.*, 2003),

– *simulated annealing* (Janzura *et al.*, 2006),

– *tabu search* (Munteanu *et al.*, 2001),

– *genetic algorithm* (Larrañaga *et al.*, 1996), *etc*.

En 2003, Blanco *et al.* (2003) puis Romero *et al.* (2004) proposent d'utiliser les **EDA** (***E**stimation of **D**istribution **A**lgorithms*) (Mülhenbein *et al.*, 1996) pour palier les problèmes rencontrés avec les algorithmes génétiques et décrits dans Larrañaga *et al.* (2000). De nombreux travaux montrent l'efficacité des **EDA** dans de nombreux domaines (Larrañaga *et al.*, 2001), y compris pour l'apprentissage de réseaux Bayésiens où les **EDA** surpassent les algorithmes génétiques (Blanco *et al.*, 2003; Romero *et al.*, 2004; Thibault *et al.*, 2007).

2.2.1. *Algorithmes EDA*

Les **EDA** sont des algorithmes « évolutionnaires » à base de populations. Ils reposent sur une connaissance explicite de la distribution des individus dans l'espace des solutions n'utilisant ainsi aucun opérateur de croisement ou mutation. Les **EDA** repartissent les individus d'une génération selon la distribution manipulée, puis réestiment cette distribution selon les meilleurs individus. Dans notre précédent travail expérimental (Thibault *et al.*, 2007) nous avons sélectionné la variante **UMDA** de l'algorithme, qui donnait de meilleurs résultats que la variante **PBIL**. Nous reprenons ici nos conclusions, en paramétrant l'algorithme avec 1000 individus, une sélection élitiste de 100 individus et sans conservation des individus d'une population sur la suivante.

Algorithm 1 Algorithme EDA (variante UMDA)

fonction EDA(M, N) ▷ M taille de la population, N taille de la sélection
  $dist \leftarrow$ distribution uniforme initiale
  répéter
    $population \leftarrow$ GÉNÈRE_POPULATION_SELON_DIST($dist, M$)
    $meilleurs \leftarrow$ SÉLECTIONNE_MEILLEURS($population, N$)
    $dist \leftarrow$ RÉESTIME_DISTRIBUTION($meilleurs$)
  jusqu'à CONVERGENCE($dist$) $== 1$
  retourne $meilleur\_individu$
fin fonction

Illustrons l'algorithme sur *One Max*, un problème jouet dont le principe est de maximiser la somme des éléments d'un vecteur binaire. La *fitness*[2] choisie ici est le nombre de 1 par ligne.

ind.	x_1	x_2	x_3	fit
$x(1)$	0	1	0	1
$x(2)$	0	1	0	1
x(3)	***1***	***0***	***1***	***2***
x(4)	***1***	***0***	***1***	***2***
x(5)	***0***	***1***	***1***	***2***
$x(6)$	1	0	0	1
$dist$	0.5	0.5	0.5	
$dist'$	***0.66***	***0.33***	***1***	

$\Rightarrow$

ind.	x_1	x_2	x_3	fit
x(1)	***1***	***1***	***1***	***3***
x(2)	***1***	***1***	***1***	***3***
x(3)	***1***	***0***	***1***	***2***
$x(4)$	1	0	1	2
$x(5)$	1	0	1	2
$x(6)$	0	0	1	1
$dist$	0.7	0.3	1	
$dist'$	***1***	***0.66***	***1***	

$\Rightarrow \ldots$

Partant d'une distribution uniforme 6 individus sont générés (1.4). A partir d'une sélection (1.5) des 3 meilleurs (en gras), la distribution est ré-estimée (1.6) en calculant la probabilité fréquentielle de chaque partie de la solution (formule 6). Ce processus recommence jusqu'à un critère d'arrêt (1.7) — généralement la convergence.

$$dist'_i = \frac{1}{N} \sum_{k \in meilleurs} x_i(k) \qquad [6]$$

2. La fonction à maximiser qui représente la capacité d'un individu à survivre et se reproduire.

Notre application de l'algorithme aux réseaux Bayésiens utilise une matrice qui associe à chaque arc $X \rightarrow Y$ une probabilité, ce qui suppose que la distribution des structures peut se calculer à partir d'une combinaison des probabilités de chaque arc. La règle de mise à jour est similaire à celle de l'équation 6. Notons qu'une telle structure nécessite l'utilisation d'un opérateur de coupure des cycles orientés.

2.2.2. *Algorithme GES*

En 2002, Chickering (2002b) publient l'algorithme **GES** qui travaille directement dans l'espace des équivalents. Pour comprendre son fonctionnement, il faut revenir sur quelques notions des réseaux Bayésiens.

Une CARTE D'INDÉPENDANCE est un graphe $\mathcal{G}$ tel que $(\mathcal{G}, P)$ respecte la condition de *Markov* (Verma *et al.*, 1988) (où P une distribution). On parle de carte d'indépendance minimale lorsqu'aucun graphe partiel de $\mathcal{G}$ n'est une carte d'indépendance de $\mathcal{P}$. A l'inverse, une CARTE DE DÉPENDANCE est un graphe $\mathcal{G}'$ tel que toute indépendance dans P implique une séparation dans $\mathcal{G}'$. Parallèlement, on peut définir la notion de carte maximale de dépendance. Une CARTE PARFAITE est l'unique carte qui soit à la fois une carte d'indépendance (minimale) et une carte de dépendance (maximale). Lorsque le nombre d'exemples est suffisamment grand, la distribution $\mathcal{P}$ de la base d'apprentissage est une carte parfaite du réseau original $\mathcal{G}$ ($\Leftrightarrow$ ($\mathcal{G}$,$\mathcal{P}$) est "*faithful*"). Notons que la séparation est une notion générale des graphoïdes (Pearl, 1988), sa version « orientée » est la d-séparation — qui s'applique donc aux réseaux Bayésiens.

Un arc couvert est un arc $X \rightarrow Y$ tel que les parents de X et Y sont identiques (à part X qui n'est pas un parent de lui-même). S'il existe une réprésentation *DAG-faithful* $\mathcal{H}$ de la distribution $\mathcal{P}$, alors le théorème de *Meek* (Meek, 1997) (théorème 1) prouve qu'il existe une suite finie d'opérations pour trouver la carte $\mathcal{H}$. Cette conjecture a été démontrée par Chickering (2002b).

Théorème 1 (rappel) *Soit $\mathcal{G}$ et $\mathcal{H}$ deux **DAG** tels que $\mathcal{H}$ est une carte d'indépendance de $\mathcal{G}$. Soit r le nombre d'arcs dans $\mathcal{H}$ qui ont une orientation opposée dans $\mathcal{G}$, et m le nombre d'arcs de $\mathcal{H}$ qui n'existent dans aucune orientation dans $\mathcal{G}$. Il existe une séquence d'au plus $r + 2m$ opérations de réversion ou d'ajout d'arc dans $\mathcal{G}$ telle que :*

– *chaque reversion concerne un arc couvert,*

– *après chaque opération, $\mathcal{G}$ est un **DAG** et $\mathcal{H}$ une carte d'indépendance de $\mathcal{G}$,*

– *après toutes les réversions ou ajouts, $\mathcal{G} = \mathcal{H}$.*

Plusieurs algorithmes utilisent le résultat de *Meek* (Meek, 1997) mais **GES** travaille dans l'espace des équivalents grâce à des opérateurs définit pour cet espace (Chickering, 2002a). Chaque classe d'équivalence est réprésentée par un **CPDAG** (***C**ompleted **P**artially **D**irected **A**cyclic **G**raph*), ou *pattern* depuis que (Verma *et al.*, 1991) ont montrés que tous les graphes d'une même classe possèdent le même squelette et les mêmes V-structures, *i.e.*, les mêmes liaisons convergentes. L'opération qui oriente d'une façon quelconque les arcs non-orientés — appelée INSTANCIATION — est close dans l'espace des équivalents.

GES procède en deux étapes : une étape gloutonne d'insertion d'arcs (dans $\mathbb{E}$) qui permet d'augmenter le graphe jusqu'à l'obtention d'une carte d'indépendance, puis une étape de suppression d'arcs permettant de minimiser la carte d'indépendance, donc de trouver la carte parfaite. L'utilisation d'un score consistant est donc nécessaire pour découvrir les opérations d'améliorations.

Algorithm 2 Algorithme GES

```
1: fonction GES
2:     G ← ∅
3:     tant que arc ← TROUVER_AMELIORATION("ajout",G) faire
4:         G ← AJOUTE_ARC(G, arc)        ▷ obtention d'une carte d'indépendance
5:     fin tant que
6:     tant que arc ← TROUVER_AMELIORATION("deletion",G) faire
7:         G ← RETIRE_ARC(G, arc)        ▷ minimisation de la carte d'indépendance
8:     fin tant que
9:     retourne G                        ▷ carte minimale d'indépendance ⇔ carte parfaite
10: fin fonction
```

3. Comparaisons

L'objet de cette étude est d'évaluer la qualité des résultats des approches par **EDA** en comparant avec l'algorithme **GES** (ou ***G**reedy **E**quivalent **S**earch*) (Chickering, 2002b), un algorithme très performant utilisant l'espace des équivalents de *Markov*. Pour cela, nous comparons les résultats obtenus sur plusieurs *benchmarks* à l'aide de plusieurs critères de qualité.

3.1. *Benchmarks*

Nous avons utilisés 5 réseaux tests classiques de la littérature : ASIA (Lauritzen *et al.*, 1988), INSURANCE (Binder *et al.*, 1997), INSULIN (Le *et al.*, 2004), ALARM (Beinlich *et al.*, 1989), HAILFINDER (Abramson *et al.*, 1996). Pour chacun, une base d'apprentissage de taille arbitraire (selon le nombre de variables et de modalités) a été générée par la méthode "*logic sampling*" (Henrion, 1988). Chaque réseau a servi à la génération de 10 bases de d'apprentissage. Les résultats sont des moyennes sur 10 lancements différents avec les écarts-types correspondants.

3.2. *Criteres de comparaison*

Tandis que les **EDA** effectuent leur recherche dans $\mathbb{B}$ — l'espace des réseaux Bayésiens — **GES** utilise $\mathbb{E}$ — l'espace des équivalents. L'utilisation d'un score possédant

benchmarks	variables	type	arcs	données
ASIA	8	binaires	8	2 000
INSURANCE	27	discrètes	52	10 000
INSULIN	35	binaires	52	5 000
ALARM	37	discrètes	46	10 000
HAILFINDER	56	discrètes	66	30 000

Tableau 1. *Réseaux tests :* ASIA, INSURANCE, INSULIN, ALARM *et* HAILFINDER

la propriété d'équivalence assure de pouvoir comparer les scores des structures obtenues par les algorithmes. Le score utilisé, DÉCOMPOSABLE, ÉQUIVALENT et CONSISTANT est **BDeu** avec :

– l'*a priori* sur les structures $p(\mathcal{B}) = \prod_{i=1}^{n} 0.001^{(r_i-1)q_i}$,

– l'*a priori* sur les paramètres $\alpha_{ijk} = \frac{10}{r_i \cdot q_i}$,

– donc, $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk} = \frac{10}{q_i}$.

Ce score est donné dans Chickering (2002b) pour l'implémentation de **GES**. Dans nos expérimentations, il est implémenté avec un cache **LRU** (***L**east **R**ecently **U**sed*) afin d'accélerer les calculs.

Afin de pouvoir comparer les structures, les **DAG** obtenus par l'**EDA** sont transformés en **CPDAG** par la méthode décrite par Chickering (1995). Ainsi, il est possible de compter le nombre de différences structurelles par rapport au représentant de la classe d'équivalence du *benchmark*. Ces différences structurelles peuvent être de trois types : arc en trop, mauvaise orientation (inversion, absence ou présence abusive d'orientation), ou arc manquant. La distance d'édition (**ED**) est la somme de ces trois mesures, c'est à dire le nombre de changements élémentaires nécessaires pour retrouver la structure originale.

Enfin, le nombre d'itérations de chaque algorithme est présenté. A chaque itération de l'**EDA** tous les individus sont scorés, ce qui n'est pas le cas dans **GES** puisque seules les modifications locales applicables à l'individu sont scorées. A titre indicatif, nous donnons le temps total d'exécution, directement proportionnel au nombre d'appels effectifs à la fonctions de score — *i.e.*, lorsque la valeur ne se trouve pas en cache.

3.3. *Résultats*

Le tableau 2 présente les écarts de score par rapport à la structure originale (*référence*). Afin de rendre compte de ces écarts, ils sont également donnés de façon relative. En terme de score, **GES** obtient de bien meilleurs résultats qu'**EDA**. Mieux, il obtient de meilleurs scores que celui du réseau original pour les 2 *benchmarks* INSULIN et HAILFINDER. Le score utilisé étant CONSISTANT, cela signifie que le réseau

		écart absolu		relatif (%)	
scores	*benchmark*	**EDA**	**GES**	**EDA**	**GES**
ASIA	-4741.8	-13.66 ± 8.45	-1.86 ± 9.66	0.29	0.04
INSURANCE	-149113	-5128 ± 1701	-3829 ± 963	3.44	2.57
INSULIN	-87253.6	-970.4 ± 376	$+152.4 \pm 127$	1.11	-0.17
ALARM	-110637	-7311 ± 1049	-2925 ± 120	6.61	2.64
HAILFINDER	-1500330	-67328 ± 10434	$+4364 \pm 92$	4.49	-0.29

Tableau 2. *Ecart absolu des scores ($ecart - score\_ref$) et écart relatif ($|\frac{ecart}{score\_ref}|$)*

obtenu par l'algorithme représente mieux la distribution que le *benchmark* lui-même. Remarquons que l'écart-type est généralement plus faible pour **GES**, ce qui montre l'aspect très stochastique des **EDA**.

Le nombre d'itérations rend compte de la faculté de l'algorithme à converger, *i.e.*, trouver une solution en un temps « acceptable ». Chaque génération d'**EDA** correspond ici à l'évaluation complète de 1000 individus (hors utilisation du cache). La méthode est donc plus gourmande que **GES** qui n'évalue en général qu'un nombre assez réduit d'opérateurs. On remarquera encore l'écart-type plus important dans les **EDA** (tableau 3). La grande différence de temps observée est une conséquence du mode

	nombre d'itérations		temps d'exécution (en s.)	
temps	**EDA**	**GES**	**EDA**	**GES**
ASIA	11.5 ± 2.55	7.0 ± 0.47	0.4 ± 0.1	0.0 ± 0.1
INSURANCE	26.9 ± 1.91	42.1 ± 1.91	103.4 ± 1.4	1.2 ± 0.1
INSULIN	28.2 ± 2.86	47.1 ± 1.20	77.3 ± 1.2	1.8 ± 0.1
ALARM	30.8 ± 1.75	42.9 ± 1.52	168.0 ± 2.3	2.3 ± 0.3
HAILFINDER	34.8 ± 4.87	65.5 ± 0.71	1794.7 ± 16.7	27.8 ± 0.5

Tableau 3. *Nombre moyen d'itérations et temps moyen d'exécution avant convergence*

de fonctionnement de chaque algorithme. Quand l'un procède par « petites » modifications locales, améliorant de fait considérablement la solution, l'autre dissémine de nombreux individus dans l'espace de recherche afin de le « fouiller ».

En général, l'objectif des méthodes à base de score n'est pas de retrouver la structure initiale, mais de trouver un modèle à la fois simple et précis en inférence. Toutefois, la CONSISTANCE de notre score permet de distinguer les structures « proches » de la structure originale. Un score consistant augmente lorsque le nombre d'indépendances conditionnelles retrouvées augmente, mais diminue si trop d'indépendances supplémentaires sont encodées dans la structure. Nous pouvons donc observer les résultats en terme de défauts structurels (tableau 4) où fp représente le nombre de faux-positifs, fn les faux-négatifs et fo les fautes-d'orientation par rapport au réseau

original[3], la somme formant la distance d'édition (ed). Le nombre d'erreurs d'orien-

erreurs		*fp*	*fn*	*fo*	*ed*
EDA	ASIA	1.5 ± 0.7	1.1 ± 0.6	4.6 ± 1.7	7.2 ± 2.1
	INSURANCE	11.3 ± 3.2	20.0 ± 1.9	23.6 ± 3.0	54.9 ± 6.1
	INSULIN	8.4 ± 4.0	13.5 ± 2.8	10.0 ± 2.1	31.9 ± 7.0
	ALARM	21.4 ± 2.8	11.5 ± 2.0	28.1 ± 2.3	61.0 ± 4.2
	HAILFFINDER	26.9 ± 4.3	33.4 ± 2.8	28.3 ± 3.1	88.6 ± 6.0
GES	ASIA	0.2 ± 0.6	1.2 ± 0.8	0.3 ± 0.7	1.7 ± 1.8
	INSURANCE	6.1 ± 1.7	16.0 ± 1.3	14.8 ± 0.8	36.9 ± 2.8
	INSULIN	1.0 ± 0.5	5.9 ± 1.1	7.5 ± 1.8	14.4 ± 2.8
	ALARM	5.1 ± 0.3	8.4 ± 0.7	4.4 ± 0.5	17.9 ± 0.3
	HAILFFINDER	15.8 ± 2.5	20.3 ± 2.1	10.5 ± 2.0	46.6 ± 4.9

Tableau 4. *Qualité de la structure obtenue avec les algorithmes **EDA** et **GES***

tation est légèrement biaisé : calculant ce critère sur les des **CPDAG**, une mauvaise orientation d'un arc entrainera souvent une mauvaise orientation de ses voisins par propagation. Malgrès cet obstacle, nous avons conservé cette méthode de comparaison, car chaque différence représente un changement de classe d'équivalence.

4. Conclusion

Dans ce travail, nous avons mis en évidence les limites des **EDA** en montrant que l'algorithme **GES** est plus performant en qualité et en temps d'exécution. Pour cela, nous avons comparé nos deux implémentations sur un ensemble de *benchmarks* connus, en utilisant les critères usuels. La démonstration de la conjecture de *Meek* offre un résultat théorique important, qui permet d'élaborer des algorithmes plus efficaces que les méthodes généralistes telles qu'**EDA**.

5. Bibliographie

Abramson B., Brown J., Edwards W., Murphy A., Winkler R. L., « Hailfinder : A Bayesian system for forecasting severe weather », *International Journal of Forecasting*, vol. 12, n° 1, p. 57-71, 1996.

Akaike H., « Statistical predictor identification », *Annals of Institute of Statistical Mathematics*, vol. 22, n° 1, p. 203-217, December, 1970.

Beinlich I. A., Suermondt H. J., Chavez R. M., Cooper G. F., « The ALARM monitoring system : a case study with two probabilistic inference techniques for belief networks », *proceedings of the Second European Conference on Artificial Intelligence in Medicine*, vol. 38, Springer-Verlag, Berlin, London, Great Britain, p. 247-256, 1989.

3. Attention, ces différences sont celles qui s'appliquent aux **PDAG** et non aux **DAG**.

Binder J., Koller D., Russell S., Kanazawa K., « Adaptive Probabilistic Networks with Hidden Variables », *Mach. Learn.*, vol. 29, n° 2-3, p. 213-244, 1997.

Blanco R., Inza I., Larrañaga P., « Learning Bayesian networks in the space of structures by estimation of distribution algorithms », *International journal of intelligent systems*, vol. 18, p. 205-220, 2003.

Bouckaert R., « Probabilistic network construction using the Minimum Description Length principle », *Lecture Notes in Computer Science*, vol. 747, p. 41-48, 1993.

Buntine W., « Theory refinement on Bayesian networks », *Proceedings of the 7th conference on Uncertainty in artificial intelligence*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, p. 52-60, 1991.

Buntine W., « A guide to the literature on learning probabilistic networks from data », *IEEE Transactions On Knowledge And Data Engineering*, vol. 8, p. 195-210, 1996.

Campos L. M. D., Fernández-Luna J. M., Puerta J. M., « An iterated local search algorithm for learning Bayesian networks with restarts based on conditional independence tests », *International Journal of Intelligent Systems*, vol. 18, n° 2, p. 221-235, 2003.

Cheng J., Bell D., Liu W., « Learning Bayesian networks from data : an information-theory based approach », *Artificial Intelligence*, vol. 137, n° 1-2, p. 43-90, 2002.

Chickering D. M., « A Transformational Characterization of Equivalent Bayesian Network Structures », *Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, p. 87-98, August, 1995.

Chickering D. M., « Learning Bayesian Networks is NP-Complete », *in* D. Fisher, H. Lenz (eds), *Learning from Data : Artificial Intelligence and Statistics*, vol. 5, Springer-Verlag, p. 121-130, 1996.

Chickering D. M., « Learning equivalence classes of Bayesian network structures », *Journal of Machine Learning Research*, vol. 2, p. 445-498, 2002a.

Chickering D. M., « Optimal structure identification with greedy search », *Journal of Machine Learning Research*, vol. 3, p. 507-554, November, 2002b.

Chickering D. M., Geiger D., Heckerman D., « learning bayesian networks : search methods and experimental results », *Proceedings of 5th Conference on artificial Intelligence and Statistics*, p. 122-128, January, 1995.

Chow C. K., Liu C. N., « Approximating discrete probability distributions with dependence trees », *IEEE Transactions on Information Theory*, vol. 14, n° 3, p. 462-467, May, 1968.

Cooper G. F., Hersovits E., « A bayesian method for the induction of probabilistic networks from data », *Journal of Machine Learning*, vol. 9, n° 4, p. 309-347, Octobre, 1992.

Friedman N., Koller D., « Being Bayesian about Network Structure », *Proceedings of the 16th Annual Conference on Uncertainty in Artificial Intelligence*, Stanford, California, p. 201-210, 2000.

Heckerman D., Geiger D., Chickering D. M., « Learning Bayesian networks : The combination of knowledge and statistical data », *Journal of Machine Learning*, vol. 20, n° 3, p. 197-243, September, 1995.

Heckerman D. M., Meek C., Cooper G., *A Bayesian approach to causal discovery*, MIT Press, Cambridge, MA, February, 1997.

Henrion M., « Propagating uncertainty in Bayesian networks by probabilistic logic sampling », *Uncertainty in artificial intelligence*, vol. 2, p. 149-164, 1988.

Janzura M., Nielsen J., « A simulated annealing-based method for learning Bayesian networks from statistical data : Research Articles », *International Journal of Intelligent Systems*, vol. 21, n° 3, p. 335-348, 2006.

Larrañaga P., Etxeberria R., Lozano J. A., Peña J. M., « Combinatorial optimisation by learning and simulation of Bayesian networks », *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, p. 343-352, 2000.

Larrañaga P., Lozano J. A., *Estimation of Distribution Algorithms. A new tool for evolutionary computation*, Kluwer Academic Publishers, 2001.

Larrañaga P., Poza M., Yurramendi Y., Murga R. H., Kuijpers C. M. H., « Structure Learning of Bayesian Networks by Genetic Algorithms : A Performance Analysis of Control Parameters », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, IEEE Computer Society, Washington, DC, USA, p. 912-926, September, 1996.

Lauritzen S. L., Spiegelhalter D. J., « Local computations with probabilities on graphical structures and their application on expert systems », *J. Royal Statistical Society B*, vol. 50, n° 2, p. 157-224, 1988.

Le P. P., Bahl A., Ungar L. H., « Using prior knowledge to improve genetic network reconstruction from microarray data », *In Silico Biology*, vol. 4, p. 335-353, 2004.

Meek C., Graphical Models : Selecting causal and statistical models, PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1997.

Munteanu P., Bendou M., « The EQ Framework for Learning Equivalence Classes of Bayesian Networks », *in* N. Cercone, T. Y. Lin, X. Wu (eds), *Proceedings of the 2001 IEEE International Conference on Data Mining*, IEEE Computer Society, Washington, DC, USA, p. 417-424, November, 2001.

Mülhenbein H., Paaß G., « From recombination of genes to the estimation of distributions I. Binary parameters », *Parallel Problem Solving from Nature*, vol. tome PPSN IV of *Lectures Notes in Computer Science*, Springer Berlin / Heidelberg, p. 178-187, 1996.

Pearl J., *Probabilistic reasoning in intelligent systems : Networks of plausible inference*, Morgan Kaufmann Publishers, San Mateo, CA, California, September, 1988.

Romero T., Larrañaga P., Sierra B., « Learning Bayesian networks in the space of orderings with estimation of distribution algorithms », *International journal of Pattern Recognition and Artificial Intelligence*, vol. 18, n° 4, p. 607-625, 2004.

Schwarz G., « Estimating the dimension of a model », *The Annals of Statistics*, vol. 6, n° 2, p. 461-464, 1978.

Thibault G., Bonnevay S., Aussem A., « Learning Bayesian network structures by estimation of distribution algorithms : An experimental analysis », *Digital Information Management*, Lyon, France, p. 127-132, October, 2007.

Verma T., Pearl J., « Causal networks : Semantics and expressiveness », *Proceedings of the 4th Workshop on Uncertainty in Artificial Intelligence*, Association for Uncertainty in Artificial Intelligence, North-Holland Publishing Co., Amsterdam, The Netherlands, p. 69-78, 1988.

Verma T., Pearl J., « Equivalence and synthesis of causal models », *in* L. K. M. Henrion, R. Shachter, J. Lemmer (eds), *Proceedings of the 6th Conference of Uncertainty in Artificial Intelligence*, Elsevier Science Inc., New York, NY, USA, p. 255-270, 1991.