Towards a More Efficient Evolutionary Induction of Bayesian Networks

Carlos Cotta[1] and Jorge Muruzábal[2]

[1] Dept. Lenguajes y Ciencias de la Computación, ETSI Informática,
University of Málaga, Campus de Teatinos, 29071 - Málaga, SPAIN
ccottap@lcc.uma.es

[2] Grupo de Estadística y Ciencias de la Decisión, ESCET,
University Rey Juan Carlos, 28933 - Móstoles, SPAIN
j.muruzabal@escet.urjc.es

Abstract. Bayesian networks (BNs) constitute a useful tool to model the joint distribution of a set of random variables of interest. This paper is concerned with the network induction problem. We propose a number of hybrid recombination operators for extracting BNs from data. These hybrid operators make use of phenotypic information in order to guide the processing of information during recombination. The performance of these new operators is analyzed with respect to that of their genotypic counterparts. It is shown that these hybrid operators provide notably improved and rather robust results. Some remarks on the future of the area are also laid out.

1 Introduction

A Bayesian Network (BN) is a graphical model postulating a joint distribution for a target set of random variables. One of the main advantageous features of this model is the fact that it provides a neat separation between qualitative and quantitative aspects of this distribution. On one hand, the qualitative aspects are given by the underlying graphical structure, a *Directed Acyclic Graph* (DAG) $\mathbf{G}$. On the other hand, quantitative aspects are provided by the set of probabilities attached to this DAG, say $\boldsymbol{\theta} = \boldsymbol{\theta}(\mathbf{G})$.

Two well-defined problems can be identified within this context: the *network induction* problem (learning an appropriate BN model), and the *inference* problem (determining the predictive conditional distribution at some variable of interest given a BN model and the values taken by certain other variables). While the latter arises when a BN has already been identified and is to be deployed in a given application, the former appears as a previous step. The focus of this work is precisely on this induction problem.

The main issue in the induction problem is learning the structure or DAG $\mathbf{G}$ (a variety of methods can be used to learn the probabilities $\boldsymbol{\theta}$). This turns out to be NP-hard [6], and hence the use of heuristic algorithms is in order [11]. In this sense, evolutionary algorithms [2] (EAs) emerge as interesting candidates for this task. Here we concentrate on the use of EAs for BN induction. More precisely, we explore in detail the role of recombination for this purpose.

The organization of the paper is as follows. Section 2 presents details of the BN framework and lays out the basic learning problem addressed later. Section 3 introduces the new operators and Section 4 reviews the empirical evidence. Finally, Section 5 closes with some discussion and prospects for future research.

2 Background

This section provides basic ideas and notational details about both BNs and some scoring metrics used for evaluation purposes. A brief overview of EA approaches for evolving DAGs is provided too.

2.1 Bayesian Networks

As mentioned above, a BN is a tuple $(\mathbf{G}, \boldsymbol{\theta})$, where $\mathbf{G}$ is a DAG and $\boldsymbol{\theta} = \boldsymbol{\theta}(\mathbf{G})$ is a set of probability distributions attached to nodes in $\mathbf{G}$. The DAG specifies a number of links or arcs among variables or nodes. If we denote the whole set of variables as $\mathbf{X} = \{X_1, X_2, ..., X_n\}$, each variable X_j has a set of parents denoted by $\Pi_j = \{X_i \in \mathbf{X} \mid \{X_i \to X_j\} \in \mathbf{G}\}$. Then, the DAG $\mathbf{G}$ represents the (skeleton) joint distribution $P(\mathbf{X}) = \prod_{i=1}^{n} P(X_i \mid \Pi_i)$. Note that at least one of the Π_i is empty; we talk of *root* nodes in this case.

A standard BN model arises when data are assumed to follow independent Multinomial distributions, that is, $P(X_i = k \mid \Pi_i = j) = \theta_{ijk}$, where $j = 1, ..., q_i$; $k = 1, ..., r_i$; r_i is the number of distinct values that X_i can assume and q_i is the number of different configurations that Π_i can present. Hence, $\boldsymbol{\theta} = \{\theta_{ijk}\}$ collects all parameters in $\mathbf{G}$ and we have $\sum_k \theta_{ijk} = 1$ for all i and j.

Given a DAG $\mathbf{G}$ and a data matrix $\mathbf{D}$ with n columns and an arbitrary number of exchangeable rows (N), the *likelihood* of the network probabilities $\boldsymbol{\theta}$ is given by the double product of the above Multinomials: $P(\mathbf{D}|\mathbf{G}, \boldsymbol{\theta}) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}}$, where N_{ijk} is the absolute frequency of value k in X_i when its parent configuration Π_i assumes state j. Given maximum likelihood estimators (MLE) $\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}}(\mathbf{G}, \mathbf{D})$ of $\boldsymbol{\theta}$, $P(\mathbf{D}|\mathbf{G}, \hat{\boldsymbol{\theta}})$ can serve as a rudimentary scoring metric. We nevertheless focus on an alternative *Bayesian* measure. Consider the *marginal* likelihood

$$P(\mathbf{D}|\mathbf{G}) = \int P(\mathbf{D}|\mathbf{G}, \boldsymbol{\theta}) \pi(\boldsymbol{\theta}|\mathbf{G}) d\boldsymbol{\theta}, \tag{1}$$

where $\pi(\boldsymbol{\theta}|\mathbf{G})$ is a prior distribution on $\boldsymbol{\theta}$. If this measure is combined with a prior distribution on DAG structures $\pi(\mathbf{G})$, the log-posterior

$$F(\mathbf{G}) = \log \pi(\mathbf{G}|\mathbf{D}) = \log \pi(\mathbf{G}) + \log P(\mathbf{D}|\mathbf{G}) \tag{2}$$

is obtained. We will be using this Bayesian measure F for some particular choices of $\pi(\mathbf{G})$ and $\pi(\boldsymbol{\theta}|\mathbf{G})$. To be precise, the former is chosen as $\pi(\mathbf{G}) \propto N^{-g/2}$, where $g = \sum_{i=1}^{n}(r_i - 1)q_i$ is the number of free $\boldsymbol{\theta}$ parameters in the model. This choice penalizes complex (i.e., highly dense) DAGs, and is closely related to the asymptotic *Bayesian Information Criterion* [10] or *BIC*[1]. As to the $\pi(\boldsymbol{\theta}|\mathbf{G})$, it is taken to be the product of independent (conjugate) Dirichlet distributions. In the case of no missing data and noninformative Dirichlet hyperparameters α, it is then possible to perform in closed form the integration leading to $P(\mathbf{D}|\mathbf{G})$ above. This key result adds to the computational tractability of the approach and will be considered here too. The hyperparameters α can be interpreted in terms of *equivalent sample size* [11]. The noninformative choice $\alpha_i = \frac{1}{r_i}$ is usually adopted following theoretical considerations related to *likelihood equivalence* [12].

2.2 Evolutionary Induction of DAGs

Typically, the EA approach for designing BNs evolves DAG structures which –when submitted for fitness calculation– are augmented with $\hat{\boldsymbol{\theta}}$ parameters and fed into a scoring function. The internal representation of DAGs turns out to be a crucial issue here. In essence, choices regarding this aspect can be classified within two main categories: *direct* and *indirect*.

Direct approaches are those in which the search is conducted over the space of all possible DAGs, say $\mathcal{S}_{DAG}$. An obvious potential problem in these approaches is the generation of infeasible solutions (i.e., digraphs with cycles). This can be avoided in two different ways. On one hand, a precedence order among variables can be assumed; then, it suffices to evolve the upper triangular portion of the adjacency matrix of the graph to obtain feasible DAGs (alternatively, closed operators in $\mathcal{S}_{DAG}$ can be defined; we will return to this point below). On the other hand, a repair function can be used to remove cycles before evaluation. See [14] for a comparison of both approaches.

As regards indirect approaches, these use an auxiliary space $\mathcal{S}_{aux}$ to conduct the search. Elements from $\mathcal{S}_{aux}$ are then fed to a suitable (decoder) algorithm to obtain the actual BNs they represent. Consider, for example, the search in the space of $n-$element permutations [13]; the construction heuristic K2 [7] is subsequently used to build the actual BN. This approach has the advantage of filtering out infeasible solutions while it also introduces problem-specific knowledge. There are some drawbacks though. For example, it may be difficult to design parsimonious operators for exploring $\mathcal{S}_{aux}$ in some situations (the $\mathcal{S}_{aux} \longrightarrow \mathcal{S}_{DAG}$ mapping can hinder the design of such operators). Also, some good regions of $\mathcal{S}_{DAG}$ may be unreachable by certain decoders.

Direct approaches do not face these hurdles as long as reproductive operators manipulate *meaningful information units*, that is, the main idea is to depart from the classical (purely syntactic) crossover in order to achieve semantically-sound recombination. This goal can be seen as an upgrading process, and hence the identification of syntactically-correct information units is still required. In this sense, a generic analysis of the syntax of these units has been done in [8]. These

[1] $BIC = \log P(\mathbf{D}|\mathbf{G}, \hat{\boldsymbol{\theta}}) - \frac{g}{2}\log N$

authors show that minimal *transmission units*[2] have the following structure:

$$\begin{aligned} T(X_i \nrightarrow X_j, \Phi) &= \{X_i \nrightarrow X_j\} \\ T(X_i \rightarrow X_j, \Phi) &= \{X_i \rightarrow X_j\} \cup \{X_r \nrightarrow X_s \mid C^{\oplus}_{sr} = 1\} \end{aligned} \tag{3}$$

where Φ is the partially-defined descendant DAG at any intermediate step of recombination, $X_i \rightarrow X_j$ (resp. $X_i \nrightarrow X_j$) represents the decision of including in (resp. excluding from) Φ the arc from X_i to X_j, $C^{\oplus} = C^{\infty}_{\Phi}$ XOR $C^{\infty}_{\Phi \cup \{X_i \rightarrow X_j\}}$, and C^{∞}_{Ψ} is the transitive closure of a graph Ψ. The next section shows how the units in Eq. (3) can be endowed with semantic information.

3 Bayesian Network Recombination

Semantically-aware operators are defined in terms of phenotypic information. This implies that recombination no longer takes place at the DAG level, but at the BN level. Two hybrid-operator templates (and phenotypic measures to be used therein) are discussed below.

3.1 Genetic vs. Allelic Recombination

Any DAG $\mathbf{G}$ can be viewed as the composition of a number of basic units η^x_{ij}, where i, j are nodes and $x \in \{1, 0\}$ indicates whether the corresponding directed arc is present in $\mathbf{G}$ or not. Informally speaking, each η_{ij} is a *gene*, whereas each η^x_{ij} is an *allele* for that gene. Two recombination approaches are thus possible.

A first possibility is to focus on individual genes. A recombination operator following this criterion must process all genes (in any suitable –not necessarily fixed– order) to construct a valid solution. For each gene, a decision must be made on whether to use the allele from either the father $\mathbf{G}$ or the mother $\mathbf{H}$[3]. While a genotypic operator would make these decisions at random, the use of phenotypic information is proposed here. More precisely, a Boolean function β –taking the two parent BNs ($\mathbf{G}$ and $\mathbf{H}$) and the partially-built child (Ξ) as input– determines the value of each gene. The pseudocode of this generic operator (termed PheGT for 'phenotypic gene transmission') is as follows:

```
1. for $i \in \{1..n\}$ do $\Pi^{\Xi}_i \leftarrow \emptyset$
2. for $i \in \{1..n\}$ do $\Upsilon_i \leftarrow (\Pi^{\mathbf{G}}_i \cup \Pi^{\mathbf{H}}_i)$
3. while $\exists \Upsilon_j \neq \emptyset$ do
   (a) Pick $X_i \in \Upsilon_j$
   (b) $\Upsilon_j \leftarrow \Upsilon_j \setminus \{X_i\}$
   (c) if $\beta(\eta_{ij}, \mathbf{G}, \mathbf{H}, \Xi)$ then
        i. $\Pi^{\Xi}_j \leftarrow \Pi^{\Xi}_j \cup \{X_i\}$
       ii. for $[X_k \nrightarrow X_s] \in T(X_i \rightarrow X_j, \Xi)$ do $\Upsilon_s \longleftarrow \Upsilon_s \setminus \{X_k\}$
```

[2] Transmission units can be seen as minimal –not necessarily elementary– pieces of information that have to transmitted as a whole from parents to offspring so as to ensure feasibility of the latter.

[3] Extensions to multiparent recombination are straightforward.

A different template arises when the emphasis is put on individual alleles. In this case, all η^1_{ij} alleles taken from either parent are put on a common bag. Then, the operator iteratively decides which alleles are extracted and injected (together with the corresponding transmission unit) into the child. The operator may decide to terminate transmission at any point along the process; all unspecified genes are given the default value η^0_{ij} in this case. Phenotypic information can be used here for both deciding the order in which alleles are picked (using a selection function σ), and for determining when to stop (using a Boolean function τ). The corresponding template of this operator (termed PheAT for 'phenotypic allele transmission') is as follows:

```
1. for $i \in \{1..n\}$ do $\Pi^{\Xi}_i \leftarrow \emptyset$
2. $\Upsilon \longleftarrow \langle \eta^1_{ij} \mid X_i \in \Pi^{\mathbf{G}}_j \cup \Pi^{\mathbf{H}}_j \rangle$
3. while $\neg\tau(\Upsilon, \Xi)$ do
   (a) $\eta^1_{ij} \leftarrow \sigma(\Upsilon, \Xi)$
   (b) $\Upsilon \leftarrow \Upsilon \setminus \{\eta^1_{ij}\}$
   (c) $\Pi^{\Xi}_j \leftarrow \Pi^{\Xi}_j \cup \{X_i\}$
   (d) for $[X_k \not\twoheadrightarrow X_s] \in T(X_i \rightarrow X_j, \Xi)$ do $\Upsilon \leftarrow \Upsilon \setminus \{\eta^1_{ks}\}$
```

Some possible instantiations of the above templates (β for PheGT; σ and τ for PheAT) are discussed next.

3.2 Phenotypic Measures

The mutual information $MI(X_j, X_i)$ criterion has often been the choice for measuring the *merit* of single alleles η^1_{ij} [16]. However, this measure is known to have some limitations due to its myopic nature [9]. The *updated* MI measure, namely the *Conditional Mutual Information* measure [9]

$$CMI(X_j, X_i \mid\mid \Pi_j \setminus \{X_i\}) = \sum P(\Pi_j \setminus \{X_i\}) \sum P(X_j, X_i \mid \Pi_j \setminus \{X_i\}) \log \frac{P(X_j, X_i \mid \Pi_j \setminus \{X_i\})}{P(X_j \mid \Pi_j \setminus \{X_i\}) P(X_i \mid \Pi_j \setminus \{X_i\})} \quad (4)$$

reflects the strength of the association between X_j and X_i once the effect of $\Pi_j \setminus \{X_i\}$ is taken into account. While this CMI measure deserves further attention, in this work we have decided to explore a somewhat simpler but nonetheless interesting variant thereof. Specifically, given that $X_i \in \Pi_j$ in either parent, we consider the grand average

$$\mu_{ij} = \frac{r_i}{r_j q_j} \sum Var(X_i, y, w), \quad (5)$$

where the sum ranges across both the $\frac{q_j}{r_i}$ different values w that $\Pi_j \setminus \{X_i\}$ can take and the r_j different values y that X_j can take. The inner term $Var(X_i, y, w)$ is defined as the *variance* of the probabilities $P(X_j = y \mid X_i = z, \Pi_j \setminus \{X_i\} = w)$ across the r_i different values z that X_i can take. These probabilities are of course nothing but $P(X_j = y \mid \Pi_j = (z, w)) = \theta_{j(z,w)y}$ with our earlier notation. As usual, these theoretical μ_{ij} are replaced in practice by their MLE $\hat{\mu}_{ij}$ based on $\hat{\boldsymbol{\theta}}$. If for some (y, w) the estimate of $Var(X_i, y, w)$ is close to 0, we conclude that any $X_i = z$ adds nothing new to what w already tells us about y. It is easy to see that Eq. (4) is also close to 0 in this case. Conversely, if $Var(X_i, y, w)$ is relatively large, then it does matter what X_i has to say in that situation, so we would tend to use both X_i and $\Pi_j \setminus \{X_i\}$ when predicting X_j (in this particular DAG and in general – recall that there is no explicit conservation law for the $\Pi_j's$ from parents to children).

This μ measure can be used within the operator templates presented earlier. We begin with β (central in PheGT). According to Eq. (5), μ_{ij} is always in [0, 0.25]. Thus, a first option is to use $\mu'_{ij} = 4\mu_{ij}$ as our transmission probability: $\beta(\eta_{ij}, \cdot) \equiv URand(0, 1) < \mu'_{ij}$. Since this can be a rather demanding criterion for arc transmission, we also consider the more relaxed $\mu''_{ij} = 2\sqrt{\mu_{ij}}$.

As regards PheAT, an allele-selection function σ is required. This admits a direct instantiation since we can always pick the allele with the highest μ_{ij} value (no rescaling required here). A simple (genotypic) criterion has been chosen in turn for the termination function τ. Specifically, a random number is first drawn from a $Binomial(\nu,\phi)$ distribution, where $\phi = 1/2$ and $\nu/2$ approximates the parents' mean number of arcs, and arc transmission is terminated as soon as the child reaches the desired number of arcs (or no transmittable arc remains). With this choice, we can compare PheAT to a pure genotypic version (picking alleles at random).

4 Experimental Results

We have tested a steady-state EA (*popsize* = 100, *maxevals* = 15000, crossover rate $p_X = .9$, mutation rate $p_m = 1/n^2$), using tournament selection (tournament size = 3). No fine tuning of these parameters was attempted. The initial population is obtained by generating DAGs at random[4]. The goal is to minimize the fitness function $-F(\mathbf{G}) = -\log P(\mathbf{D}|\mathbf{G}) + \frac{g}{2}\log N$, see Eq. (2) above.

Two networks have been chosen to benchmark the proposed approach: the ALARM network, a 37-variable network for monitoring patients in the intensive care unit [3], and the INSURANCE network, a 27-variable BN for evaluating car insurance risks [4]. However, due to space constraints, we concentrate here on the former (similar qualitative results have been obtained with the latter). Training sets of $N = 2,000$ examples were simulated from the ALARM network.

We have tried both the phenotypic (PheGT and PheAT) as well as the genotypic (GT and AT) operators. Two variants of each operator have been considered in turn: respectful and non-respectful. The property of *respect* [15] refers in this case to the *initial* transmission of all arcs shared by the parent DAGs. Since acyclicity is enforced at all times, inclusion of (some of) these arcs may be impossible later[5]. Hence, this initial transmission introduces an important

[4] The size of the sets Π_j is limited by $q_j < 2^{11}$.

[5] For example consider DAGs $\mathbf{G}$ and $\mathbf{H}$ such that $\{X_i \rightarrow X_j, X_j \rightarrow X_k\} \in \mathbf{G}$, and $\{X_k \rightarrow X_i, X_i \rightarrow X_j\} \in \mathbf{H}$. If arcs $\{X_j \rightarrow X_k, X_k \rightarrow X_i\}$ are transmitted to the child, it will be impossible to transmit the common arc $X_i \rightarrow X_j$ as well.

Table 1. Results of the different crossover operators on the ALARM network (averaged for 10 runs).

	$-F^*$		$-\log P(\mathbf{D}\|\mathbf{G})$	
Operator	best	mean ± std.dev	best	mean ± std.dev.
GT	28718.18	29888.03 ± 584.18	26931.97	28045.57 ± 631.24
AT	29470.42	29901.69 ± 311.24	27338.98	28085.83 ± 382.90
PheGT	29314.62	30038.57 ± 528.11	27646.22	28614.16 ± 567.53
PheAT	25596.55	26115.35 ± 469.76	24106.94	24726.66 ± 534.44
GT^R	24290.02	24807.84 ± 328.75	22617.82	23111.70 ± 285.17
AT^R	24490.63	24896.07 ± 269.84	22806.67	23139.50 ± 224.46
PheGT^R	23929.07	24493.83 ± 317.99	22291.76	22891.72 ± 368.90
PheAT^R	23861.68	24430.92 ± 379.92	22216.06	22729.01 ± 299.41
PheGT$_2^R$	23944.63	24245.57 ± 149.02	22422.18	22671.80 ± 170.73
HC	24528.41	24732.48 ± 168.53	22907.42	23000.66 ± 86.04
ALARM	24922.33		22987.90	

qualitative change in behavior. As regards the μ'_{ij} vs. μ''_{ij} choice in PheGT, we consider only the respectful variant and mark the latter option with a subscript. For comparative purposes, a hill climbing (HC) algorithm has also been tested. This HC performs single-arc insertions and deletions and has been run for the same number of evaluations as the EAs (re-start was performed each time stagnation was reached). Table 1 shows the results.

A quick inspection of these results leads to several conclusions of interest. Note first that the introduction of respect yields a substantial improvement in all performance measures. It can also be seen that the phenotypic operators clearly outperform[6] their genotypic counterparts (thus confirming the usefulness of the phenotypic approach), whereas the HC algorithm lies somewhere in between. Additionally, the networks provided by PheAT and PheGT are definitely better (in terms of the selected measures) than the original network. This feature is due to the small size of the training set and indicates that our best operators achieve some refinement in the BNs they produce. Finally, the non-linear mapping leading to μ''_{ij} in PheGT$_2^R$ provides the best overall results. Clearly, this option boosts the ability of PheGT for exploring and finding improved structures.

The structural properties of the evolved BNs are consistent with the analysis above. Table 2 shows the total number of $\boldsymbol{\theta}$ parameters (g) as well as the BIC measure discussed earlier. It can be seen that the networks provided by AT and PheAT are slightly more complex on average than those produced by GT and PheGT, whereas all of them tend to be simpler than the true network. Note also that the phenotypic crossover operators manage to produce networks of similar BIC than the original ALARM network; moreover, in some cases they interestingly provide even lower values.

[6] Significantly, using a standard (two-sample) t-test. The same holds when using a test set different from the training set, so as to evaluate overfitting.

Table 2. Structural properties of the networks evolved by the different crossover operators for the ALARM benchmark (averaged for 10 runs).

	#parameters			*BIC*	
Operator	min	mean ± std.dev.	max	best	mean ± std.dev.
GT	415	484.8 ± 65.29	659	28613.96	29779.14 ± 583.73
AT	375	477.8 ± 81.59	629	29372.47	29795.73 ± 308.49
PheGT	266	374.8 ± 62.55	488	29210.28	29946.81 ± 530.89
PheAT	307	365.4 ± 58.89	507	25511.70	26032.53 ± 466.78
GT^R	376	446.3 ± 38.17	505	24194.88	24708.69 ± 324.30
AT^R	425	462.2 ± 30.74	536	24396.27	24796.51 ± 266.89
PheGT^R	386	429.2 ± 27.65	483	23842.55	24433.58 ± 329.83
PheAT^R	387	451.2 ± 44.28	514	23730.24	24350.72 ± 418.22
PheGT_2^R	381	414.1 ± 20.78	441	23859.39	24155.89 ± 147.98
HC	403	455.7 ± 37.12	518	24434.77	24635.45 ± 164.56
ALARM	509			24049.43	

5 Summary and the Likely Future

We have described and evaluated several new recombination operators for evolving BNs. These operators are based on phenotypic information and thus depart from previously proposed genotypic crossover and phenotypic mutation. It has been shown that our phenotypic variants produce satisfactory results in problems of moderate complexity. In this sense, the observance of the property of respect has revealed itself as a crucial factor for the performance of these operators.

Perhaps the most challenging line of research in the wider BN induction problem refers to the possibility of performing the search over the space of BN *equivalence classes*, say $\mathcal{S}_{Eq}$ (rather than $\mathcal{S}_{DAG}$ as above). Two BNs are (Markov) equivalent if they encode the same statistical model, that is, the same set of independence and conditional independence statements. Let $[\mathbf{G}]$ denote the equivalence class of a DAG $\mathbf{G}$. Given training data generated by $\mathbf{G}$, many DAG-based algorithms use scoring measures that indeed score equally all members of $[\mathbf{G}]$. Hence, all that can be reasonably asked in this case is to reach some DAG in $[\mathbf{G}]$: these algorithms can not be expected to reconstruct $\mathbf{G}$ exactly. Note that the marginal likelihood $P(\mathbf{D}|\mathbf{G})$ in Eq. (1) is one of such metrics, yet our fitness measure $F(\mathbf{G})$ –dependent also on g– is not. It is still interesting to imagine how such an alternative search process could be carried out by an EA similar to the above. For one thing, search strategies that spend most of their time within the same equivalence class would seem rather inefficient (since they inadvertedly keep proposing the same model). We conclude by briefly providing some preliminary insights on this matter.

It turns out that equivalence classes can be compactly represented by (certain class of) *partially* directed acyclic graphs or PDAGs [1, 5]. PDAGs include directed as well as *undirected* arcs. Chickering [5] provides an algorithm that takes a given DAG $\mathbf{G}$ and outputs the PDAG $\bar{\mathbf{G}}$ that uniquely represents its equivalence class $[\mathbf{G}]$. Since $\bar{\mathbf{G}}$ and $\mathbf{G}$ have the same connectivity pattern (ig-

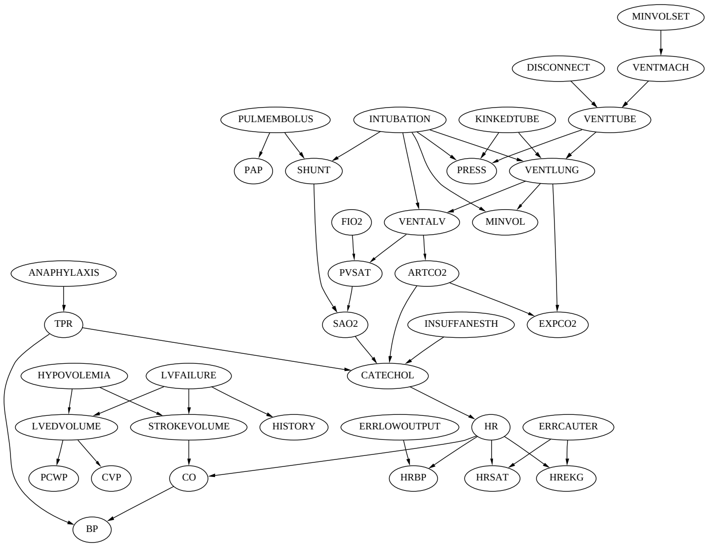

Fig. 1. The ALARM network. The reversible arcs are: MINVOLSET→VENTMACH, PULMEMBOLUS→PAP, ANAPHYLAXIS→TPR and LVFAILURE→HISTORY. We can visualize the key PDAG representing [ALARM] by making these arcs undirected.

noring directionality), all undirected arcs in $\bar{\mathbf{G}}$ correspond to *reversible* arcs in $\mathbf{G}$, whereas all directed arcs in $\bar{\mathbf{G}}$ are *compelled*: they show up throughtout [$\mathbf{G}$]. A reasonable assessment of BN quality would require correct directionality for compelled arcs but just connectivity for reversible arcs. In the ALARM network, for example, we find 4 reversible and 42 compelled arcs, see Figure 1.

To conclude, consider now potential mutation and crossover operators for some parent PDAGs $\bar{\mathbf{G}}$ and $\bar{\mathbf{H}}$. A first issue refers to PDAG *validity*: not all PDAGs represent equivalence classes. Chickering [5] presents various operators designed to modify a given $\bar{\mathbf{G}}$ so that the resulting PDAG effectively represents a *different* equivalence class. For example, both directed and undirected arcs can be added or deleted (compelled arcs can sometimes be reversed also). The familiar mutation operators found in the evolutionary DAG arena (e.g., [16]) can thus be extended along this way.

As regards crossover, an obvious approach would randomly instantiate $\bar{\mathbf{G}}$ and $\bar{\mathbf{H}}$ so as to obtain DAGs $\mathbf{G}$ and $\mathbf{H}$ from which phenotypic measures could be derived as above. The main challenge remains about how to meaningfully incorporate this DAG-based information when defining the offspring $\bar{\mathbf{K}}$ derived from $\bar{\mathbf{G}}$ and $\bar{\mathbf{H}}$. We are currently exploring some ideas in this direction.

Acknowledgement

The authors are partially supported by grants TIC99-0754-C03-03 and TIC2001-0175-C03-03 from the Spanish CICYT agency. We appreciate the assistance of David Chickering in running his LABEL-EDGES algorithm.

References

1. S.A. Andersson, D. Madigan, and M.D. Perlman. A characterization of markov equivalence classes for acyclic digraphs. *Annals of Statistics*, 25:505–541, 1997.
2. Th. Bäck, D.B. Fogel, and Z. Michalewicz. *Handbook of Evolutionary Computation.* Oxford University Press, New York NY, 1997.
3. I.A. Beinlich, H.J. Suermondt, R.M. Chavez, and G.F. Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In J. Hunter, J. Cookson, and J. Wyatt, editors, *Proceedings of the Second European Conference on Artificial Intelligence and Medicine*, pages 247–256, Berlin, 1989. Springer-Verlag.
4. J. Binder, D. Koller, S. Russell, and K. Kanazawa. Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29:213–244, 1997.
5. D.M. Chickering. Learning equivalence classes of bayesian-network structures. Submitted manuscript, 2001.
6. D.M. Chickering, D. Geiger, and D. Heckermann. Learning bayesian networks is NP-complete. In D. Fisher and H.-J. Lenz, editors, *Learning from data: AI and Statistics V*, pages 121–130, New York NY, 1996. Springer-Verlag.
7. G. Cooper and E. Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.
8. C. Cotta and J.M. Troya. Analyzing directed acyclic graph recombination. In B. Reusch, editor, *Computational Intelligence: Theory and Applications*, volume 2206 of *Lecture Notes in Computer Science*, pages 739–748. Springer-Verlag, Berlin Heidelberg, 2001.
9. N. Friedman, I. Nachman, and D. Pe'er. Learning bayesian network structures from massive datasets: The sparse candidate algorithm. In H. Dubios and K. Laskey, editors, *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 206–215, San Francisco CA, 1999. Morgan Kaufmann.
10. D. Geiger, D. Heckerman, and C. Meek. Asymptotic model selection for directed networks with hidden variables. In E. Horvitz and F.V. Jensen, editors, *Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence*, pages 283–290, San Francisco CA, 1996. Morgan Kaufmann.
11. D. Heckerman. A tutorial on learning with bayesian networks. In M.I. Jordan, editor, *Learning in Graphical Models*, pages 301–354. Kluwer, Dordrecht, 1998.
12. D. Heckerman, D. Geiger, and D.M. Chickering. Learning bayesian networks: the combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.
13. P. Larrañaga, C.M.H. Kuijpers, R.H. Murga, and Y. Yurramendi. Learning bayesian network structures by searching for the best ordering with genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 26(4):487–493, 1996.
14. P. Larrañaga, M. Poza, Y. Yurramendi, R.H. Murga, and C.M. H. Kuijpers. Structure learning of bayesian networks by genetic algorithms: A performance analysis of control parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(9):912–926, 1996.
15. N.J. Radcliffe. Equivalence class analysis of genetic algorithms. *Complex Systems*, 5:183–205, 1991.
16. M.L. Wong, W. Lam, and K.S. Leung. Using evolutionary programming and minimum description length principle for data mining of bayesian networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(2):174–178, 1999.