

# Introducing Darwinian Networks

**Cory J. Butz**

University of Regina  
 Regina, S4S 0A2, Canada  
 butz@cs.uregina.ca

<http://www.darwiniannetworks.com>

## Abstract

*Darwinian networks* (DNs) are introduced to simplify and clarify working with Bayesian networks (BNs). Rather than modelling the variables in a problem domain, DNs represent the probability tables in the model. The graphical manipulation of the tables then takes on a biological feel. It is shown how DNs can unify modeling and reasoning tasks into a single platform.

## Introduction

This invited paper draws from (Butz, Oliveira, and dos Santos 2015a). Many different platforms, techniques and concepts can be employed while modeling and reasoning with *Bayesian networks* (BNs) (Pearl 1988). A problem domain is modeled initially as a *directed acyclic graph* (DAG), denoted  $\mathcal{B}$ , and the strengths of relationships are quantified by *conditional probability tables* (CPTs). Independencies are tested in  $\mathcal{B}$  using *d-separation* (Pearl 1988) or *m-separation* (Lauritzen et al. 1990; Zhang and Poole 1994). Reasoning with a BN can be done using  $\mathcal{B}$ , including inference algorithms such as *variable elimination* (VE) (Zhang and Poole 1994) and *arc-reversal* (AR) (Olmsted 1983), or with a secondary structure called a join tree and denoted  $\mathcal{T}$ , as in *lazy propagation* (LP) (Madsen and Jensen 1999). Considering exact inference in discrete BNs, a common task, called belief update, is to compute posterior probabilities given evidence (observed values of variables). Before performing number crunching, two kinds of variables can safely be removed, namely, *barren* variables (Zhang and Poole 1994) and what we call *independent given evidence* variables (Madsen and Jensen 1999; Zhang and Poole 1994). LP and VE treat the removal of these variables as separate steps. Furthermore, LP and VE involve multiple platforms. LP conducts inference on  $\mathcal{T}$  and test independencies in  $\mathcal{B}$ . VE first prunes barren variables from a DAG  $\mathcal{B}$ , giving a sub-DAG  $\mathcal{B}^s$ , and then prunes independent by evidence variables from the *moralization* (Pearl 1988) of  $\mathcal{B}^s$ , denoted  $\mathcal{B}_m^s$ . VE can also use  $\mathcal{B}_m^s$  to determine an *elimination ordering*, denoted  $\sigma$  (Koller and Friedman 2009), using the *min-neighbors*, *min-weight*, *min-fill*, and *weighed-min-fill* (Koller and Friedman 2009)

heuristic. By adapting a few well-known concepts in biology (Dawkins 1976), all of the above can be unified into one platform to be denoted  $\mathcal{D}$ .

*Darwinian networks* (DNs) (Butz, Oliveira, and dos Santos 2015a) are put forth as a biological perspective of BNs. A CPT  $P(X|Y)$  is viewed as a population  $p(X, Y)$  with combative traits  $X$  and docile traits  $Y$ . More generally, a BN is seen as a set of populations. In DN, how populations adapt to the deletion of other populations corresponds precisely with testing independencies in BNs. Once abstract concepts like merge and replication are used to represent multiplication, division, and addition, it follows that DN can represent VE, AR, and LP. Good elimination orderings, which are of practical importance, can be computed in DN. Besides providing a single platform for testing independencies, performing inference, and determining good elimination orderings using min-neighbors, min-weight, min-fill, and weighed-min-fill, we show how DN simplify d-separation, m-separation, VE, AR, and LP. Omitted proofs are given in (Butz, Oliveira, and dos Santos 2015c).

## Bayesian Networks

Let  $U = \{v_1, v_2, \dots, v_n\}$  be a finite set of variables, each with a finite domain, and  $V$  be the domain of  $U$ . Let  $\mathcal{B}$  denote a *directed acyclic graph* (DAG) on  $U$ . A *directed path* from  $v_1$  to  $v_k$  is a sequence  $v_1, v_2, \dots, v_k$  with arcs  $(v_i, v_{i+1})$  in  $\mathcal{B}$ ,  $i = 1, 2, \dots, k - 1$ . For each  $v_i \in U$ , the *ancestors* of  $v_i$ , denoted  $An(v_i)$ , are those variables having a directed path to  $v_i$ , while the *descendants* of  $v_i$ , denoted  $De(v_i)$ , are those variables to which  $v_i$  has a directed path. For a set  $X \subseteq U$ , we define  $An(X)$  and  $De(X)$  in the obvious way. The *children*  $Ch(v_i)$  and *parents*  $Pa(v_i)$  of  $v_i$  are those  $v_j$  such that  $(v_i, v_j) \in \mathcal{B}$  and  $(v_j, v_i) \in \mathcal{B}$ , respectively. An *undirected path* in a DAG is a path ignoring directions. A *path* in an undirected graph is defined similarly. A singleton set  $\{v\}$  may be written as  $v$ ,  $\{v_1, v_2, \dots, v_n\}$  as  $v_1 v_2 \dots v_n$ , and  $X \cup Y$  as  $XY$ .

*D-separation* (Pearl 1988) tests independencies in DAGs and can be presented as follows (Darwiche 2009). Let  $X, Y$ , and  $Z$  be pairwise disjoint sets of variables in a DAG  $\mathcal{B}$ . We say  $X$  and  $Z$  are *d-separated* by  $Y$ , denoted  $I_{\mathcal{B}}(X, Y, Z)$ , if at least one valve on every undirected path between  $X$  and  $Z$  is closed. There are three kinds of valves  $v$ : (i) a *sequential* valve means  $v$  is a parent of one of its neighbors and a child

of the other; (ii) a *divergent* valve is when  $v$  is a parent of both neighbors; and, (iii) a *convergent* valve is when  $v$  is a child of both neighbors. A valve is *open* or *closed*. A sequential or divergent valve is *closed*, if  $v \in Y$ . A convergent valve is *closed*, if  $(v \cup De(v)) \cap Y = \emptyset$ . For example, suppose  $X = a$ ,  $Y = c$ , and  $Z = f$  in DAG  $\mathcal{B}$  depicted in Figure 1 (i). To test  $I_{\mathcal{B}}(a, c, f)$  there are two undirected paths from  $a$  to  $f$ . On the path  $(a, c)$ ,  $(c, e)$ ,  $(e, f)$ , valve  $c$  is closed, since  $c$  is a sequential valve and  $c \in Y$ . Valve  $d$  is closed on the other path, since  $d$  is a convergent valve and  $\{d, h\} \cap Y = \emptyset$ . As both paths from  $a$  to  $f$  have a closed valve,  $I_{\mathcal{B}}(a, c, f)$  holds.

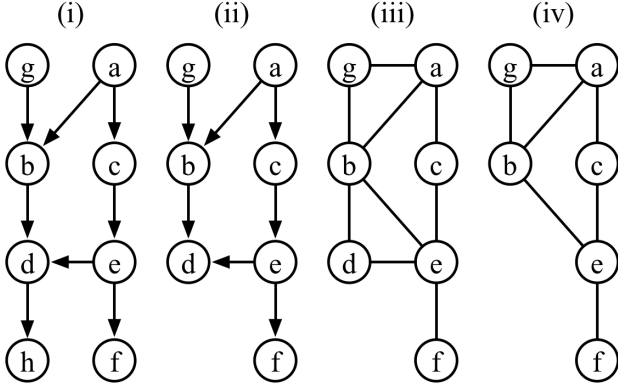


Figure 1: (i) A DAG  $\mathcal{B}$ . (ii) Sub-DAG  $\mathcal{B}^s$ . (iii) Moralization  $\mathcal{B}_m^s$ . (iv)  $\mathcal{B}_m^s$  with  $d$  and its edges deleted.

*M-separation* (Lauritzen et al. 1990; Zhang and Poole 1994) is another method for testing independencies in DAGs, and is equivalent to d-separation. Let  $X$ ,  $Y$ , and  $Z$  be pairwise disjoint sets of variables in a DAG  $\mathcal{B}$ . Then m-separation tests  $I_{\mathcal{B}}(X, Y, Z)$  with four steps: (i) construct the sub-DAG of  $\mathcal{B}$  onto  $XYZ \cup An(XYZ)$ , yielding  $\mathcal{B}^s$ ; (ii) construct the *moral graph* (Pearl 1988) of  $\mathcal{B}^s$ , denoted  $\mathcal{B}_m^s$ , by adding an undirected edge between each pair of parents of a common child and then dropping directionality; (iii) delete  $Y$  and its incident edges; and (iv) if there exists a path from any variable in  $X$  to any variable in  $Z$ , then  $I_{\mathcal{B}}(X, Y, Z)$  does not hold; otherwise,  $I_{\mathcal{B}}(X, Y, Z)$  holds. For example, in Figure 1, to test  $I_{\mathcal{B}}(a, d, f)$  in  $\mathcal{B}$  of (i), the sub-DAG  $\mathcal{B}^s$  is in (ii).  $\mathcal{B}_m^s$  is shown in (iii). Removing  $d$  and incident edges gives (iv). Since there exists a path from  $a$  to  $f$ ,  $I_{\mathcal{B}}(a, d, f)$  does not hold.

A *potential* on  $V$  is a function  $\phi$  such that  $\phi(v) \geq 0$  for each  $v \in V$ , and at least one  $\phi(v) > 0$ . A *uniform* potential on  $V$  is a function  $1$  that sets  $1(v) = 1/k$ , where  $v \in V$ ,  $k = |V|$  and  $|\cdot|$  denotes set cardinality. Henceforth, we say  $\phi$  is on  $U$  instead of  $V$ . A *joint probability distribution* is a potential  $P$  on  $U$ , denoted  $P(U)$ , that sums to one. For disjoint  $X, Y \subseteq U$ , a *conditional probability table* (CPT)  $P(X|Y)$  is a potential over  $XY$  that sums to one for each value  $y$  of  $Y$ .

A *Bayesian network* (BN) (Pearl 1988) is a DAG  $\mathcal{B}$  on  $U$  together with CPTs  $P(v_1|Pa(v_1))$ ,  $P(v_2|Pa(v_2))$ ,  $\dots$ ,  $P(v_n|Pa(v_n))$ . For example, Figure 2 (i) shows a BN,

where CPTs  $P(a)$ ,  $P(b|a)$ ,  $\dots$ ,  $P(g|e, f)$  are not shown.

We call  $\mathcal{B}$  a BN, if no confusion arises. The product of the CPTs for  $\mathcal{B}$  on  $U$  is a joint probability distribution  $P(U)$ . The *conditional independence* (Pearl 1988) of  $X$  and  $Z$  given  $Y$  holding in  $P(U)$  is denoted  $I(X, Y, Z)$ . It is known that if  $I_{\mathcal{B}}(X, Y, Z)$  holds by d-separation (or m-separation) in  $\mathcal{B}$ , then  $I(X, Y, Z)$  holds in  $P(U)$ .

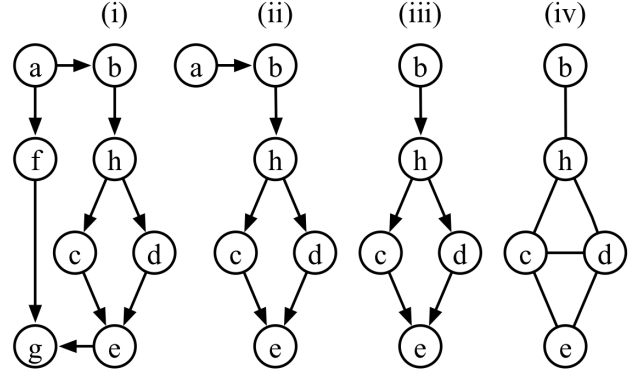


Figure 2: (Zhang and Poole 1994) Given query  $P(e|b = 0)$  posed to BN  $\mathcal{B}$  in (i), pruning barren variables  $g$  and  $f$  in (ii) and independent by evidence variable  $a$  in (iii). (iv) is  $\mathcal{B}_m^s$ .

## Inference On A Dag

*Variable elimination* (VE) (Zhang and Poole 1994) computes  $P(X|Y = y)$  from a BN  $\mathcal{B}$  as follows: (i) all barren variables are removed recursively, where  $v$  is *barren* (Zhang and Poole 1994), if  $Ch(v) = \emptyset$  and  $v \notin XY$ ; (ii) all independent by evidence variables are removed, giving  $\mathcal{B}^s$ , where  $v$  is an *independent by evidence* variable, if  $I(v, Y, X)$  holds in  $\mathcal{B}$  by m-separation; (iii) build a uniform distribution  $1(v)$  for any root of  $\mathcal{B}^s$  that is not a root of  $\mathcal{B}$ ; (iv) set  $Y$  to  $Y = y$  in the CPTs of  $\mathcal{B}^s$ ; (v) determine an elimination ordering  $\sigma$  from the moral graph  $\mathcal{B}_m^s$ ; (vi) following  $\sigma$ , eliminate variable  $v$  by multiplying together all potentials involving  $v$ , and then summing  $v$  out of the product; and, (vii) multiply together all remaining potentials and normalize to obtain  $P(X|Y = y)$ . For example (Zhang and Poole 1994), in Figure 2, given  $P(e|b = 0)$  and BN  $\mathcal{B}$  in (i),  $g$  and  $f$  are barren (ii) and  $a$  is independent by evidence (iii) for steps (i) and (ii). In steps (iii) and (iv), VE builds  $1(b)$  and updates  $P(h|b)$  as  $P(h|b = 0)$ . Step (v) can determine  $\sigma = (c, d, h)$  from  $\mathcal{B}_m^s$  shown in (iv). Step (vi) computes (step (vii) is discussed later):

$$P(c, e|d, h) = P(c|h) \cdot P(e|c, d), \quad (1)$$

$$P(e|d, h) = \sum_c P(c, e|d, h), \quad (2)$$

$$P(e|h) = \sum_d P(d|h) \cdot P(e|d, h), \quad (3)$$

$$P(e|b = 0) = \sum_h P(h|b = 0) \cdot P(e|h). \quad (4)$$

## Darwinian Networks

We foreshadow the introduction of DNs with three examples illustrated in Figure 3. The CPT  $P(e|c, d)$  is represented as a population  $p(e, cd)$  in (i). The multiplication in (2) of CPTs  $P(c|h)$  and  $P(e|c, d)$  yielding CPT  $P(c, e|d, h)$  is represented as the merge of populations  $p(c, h)$  and  $p(e, cd)$  yielding population  $p(c, e|d, h)$  in (ii). And, the marginalization in (3) of variable  $c$  from CPT  $P(c, e|d, h)$  giving CPT  $P(e|d, h)$  is represented as the replication of population  $p(ce, dh)$  giving itself and population  $p(e, dh)$  in (iii).

Adaptation and evolution are used to represent the testing of independencies and inference, respectively.

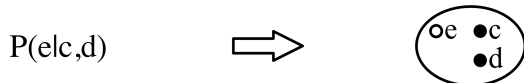
### Adaptation

A trait  $t$  can be combative or docile. A combative trait  $t_c$  is depicted by a clear (white) circle. A docile trait  $t_d$  is illustrated by a dark (black) circle. A population  $p(C, D)$  contains a non-empty set  $CD$  of traits, where  $C$  and  $D$  are disjoint,  $C$  is exclusively combative, and  $D$  is exclusively docile. A population is depicted by a closed curve around its traits. For example, Figure 4 (i) shows eight populations, including  $p(b, ag)$ , short for  $p(\{b\}, \{a, g\})$ , illustrated with a closed curve around the (clear) combative trait  $b$  and two (dark) docile traits  $a$  and  $g$ .

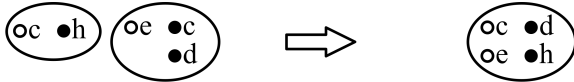
**Definition 1.** A Darwinian network (DN), denoted  $\mathcal{D}$ , is a finite, multiset of populations.

A DN  $\mathcal{D}$  is depicted by a dashed closed curve around its populations. For example, Figure 4 (i) depicts a DN  $\mathcal{D} = \{p(a), p(b, ag), p(c, a), p(d, be), p(e, c), p(f, e), p(g), p(h, d)\}$ , where  $p(C, \emptyset)$  is succinctly written  $p(C)$ .

All combative traits in a given DN  $\mathcal{D}$  are defined as  $T_c(\mathcal{D}) = \{t_c \mid t_c \in C, \text{ for at least one } p(C, D) \in \mathcal{D}\}$ . All docile traits in  $\mathcal{D}$ , denoted  $T_d(\mathcal{D})$ , are defined similarly. For example, considering DN  $\mathcal{D}$  in Figure 4 (i), then  $T_c(\mathcal{D}) = \{a_c, b_c, c_c, d_c, e_c, f_c, g_c, h_c\}$ . In addition,  $T_d(\mathcal{D}) = \{a_d, b_d, c_d, d_d, e_d, g_d\}$ .



(i) Representing CPT  $P(e|c, d)$  as population  $p(e, cd)$ .



(ii) Merging populations  $p(c, h)$  and  $p(e, cd)$  yields  $p(ce, dh)$ .



(iii) Replicating  $p(ce, dh)$  can give itself and  $p(e, dh)$ .

Figure 3: Representing a CPT from a BN as a population in a DN in (i). In (ii), merging populations corresponds to multiplying CPTs, while replicating a population corresponds to summing variables out of a CPT in (iii).

Populations are classified based upon characteristics of their traits. For adaptation, barren populations need only to be classified. Later, for evolution, we will extend the classification.

Given two DNs  $\mathcal{D}$  and  $\mathcal{D}'$ , let  $t_c$  be a trait in  $T_c(\mathcal{D})$ . Trait  $t_c$  is *strong*, if  $t_c \in T_c(\mathcal{D}')$ ; otherwise,  $t_c$  is *weak*. Trait  $t_c$  is *relict*, if  $t_d \notin T_d(\mathcal{D})$ . The notions of strong, weak, and relict are defined analogously for  $t_d$ .

Given DNs  $\mathcal{D}$  and  $\mathcal{D}'$ , a population  $p(t_c, D)$  is *barren*, if  $t_c$  is relict, and both  $t_c$  and  $t_d$  are weak.

In adaptation, *natural selection* removes recursively all barren populations from a DN  $\mathcal{D}$  with respect to a DN  $\mathcal{D}'$ .

**Example 1.** Referring to Figure 4, let us apply natural selection on the DN  $\mathcal{D}$  in (i) with respect to DN  $\mathcal{D}'$  in (v). First, barren population  $p(h, d)$  is removed. Population  $p(d, be)$  now is barren, since  $d_c$  is relict, and  $d_c$  and  $d_d$  are weak. Natural selection removes  $p(d, be)$  and, in turn,  $p(b, ag)$  and  $p(g)$ , giving (ii).

*Docilization* of a DN  $\mathcal{D}$  adds  $p(\emptyset, D)$  to  $\mathcal{D}$ , for every population  $p(C, D)$  in  $\mathcal{D}$  with  $|D| > 1$ . For example, the docilization of Figure 4 (ii) is itself, while the docilization of Figure 4 (vi) adds populations  $p(\emptyset, ag)$  and  $p(\emptyset, be)$ , giving Figure 4 (vii).

To *delete* a population  $p(C, D)$  from a DN  $\mathcal{D}$  is to remove all occurrences of it from  $\mathcal{D}$ . For example, the deletion of  $p(c, a)$  from Figure 4 (ii) gives Figure 4 (iii).

Two populations *merge* together as follows: for each trait  $t$  appearing in either population, if  $t$  is combative in exactly one of the two populations, then  $t$  is combative in the merged population; otherwise,  $t$  is docile. Let  $\mathcal{P}_X, \mathcal{P}_Y$ , and  $\mathcal{P}_Z$  be pairwise disjoint subsets of populations in a DN  $\mathcal{D}$  and let DN  $\mathcal{D}' = p(C)$ , where  $C = T_c(\mathcal{P}_X \mathcal{P}_Y \mathcal{P}_Z)$ . We test the *adaptation* of  $\mathcal{P}_X$  and  $\mathcal{P}_Z$  given  $\mathcal{P}_Y$ , denoted  $A(\mathcal{P}_X, \mathcal{P}_Y, \mathcal{P}_Z)$ , in  $\mathcal{D}$  with four simple steps: (i) let natural selection act on  $\mathcal{D}$  with respect to  $\mathcal{D}'$ , giving  $\mathcal{D}^s$ ; (ii) construct the docilization of  $\mathcal{D}^s$ , giving  $\mathcal{D}_m^s$ ; (iii) delete  $p(C, D)$  from  $\mathcal{D}_m^s$ , for each  $p(C, D)$  in  $\mathcal{P}_Y$ ; and, (iv) after recursively merging populations sharing a common trait, if there exists a population containing both a combative trait in  $T_c(\mathcal{P}_X)$  and a combative trait in  $T_c(\mathcal{P}_Z)$ , then  $A(\mathcal{P}_X, \mathcal{P}_Y, \mathcal{P}_Z)$  fails; otherwise,  $A(\mathcal{P}_X, \mathcal{P}_Y, \mathcal{P}_Z)$  succeeds.

**Example 2.** Let us test  $A(p(a), p(c, a), p(f, e))$  in the DN  $\mathcal{D}$  of Figure 4 (i), where  $\mathcal{P}_X = p(a)$ ,  $\mathcal{P}_Y = p(c, a)$ , and  $\mathcal{P}_Z = p(f, e)$ . As  $T_c(\{p(a), p(c, a), p(f, e)\}) = \{a_c, c_c, f_c\}$ , we obtain the DN  $\mathcal{D}' = p(acf)$  in Figure 4 (v). In step (i), natural selection gives  $\mathcal{D}^s$  in Figure 4 (ii). In step (ii), docilization of  $\mathcal{D}^s$  gives  $\mathcal{D}_m^s$  in Figure 4 (ii). In step (iii), the deletion of  $p(c, a)$  from  $\mathcal{D}_m^s$  gives Figure 4 (iii). Recursively merging populations in step (iv) yields Figure 4 (iv). As no population in Figure 4 (iv) contains  $a_c$  in  $T_c(p(a))$  and  $f_c$  in  $T_c(p(f, e))$ ,  $A(p(a), p(c, a), p(f, e))$  succeeds.

**Example 3.** Let us now test  $A(p(a), p(d, be), p(f, e))$  in the DN  $\mathcal{D}$  of Figure 4 (i). In this case, DN  $\mathcal{D}' = p(adf)$  is shown in Figure 4 (x). In step (i), natural selection removes barren population  $p(h, d)$  as shown in Figure 4 (vi). In step (ii), docilization of Figure 4 (vi) gives Figure 4 (vii). In step (iii),

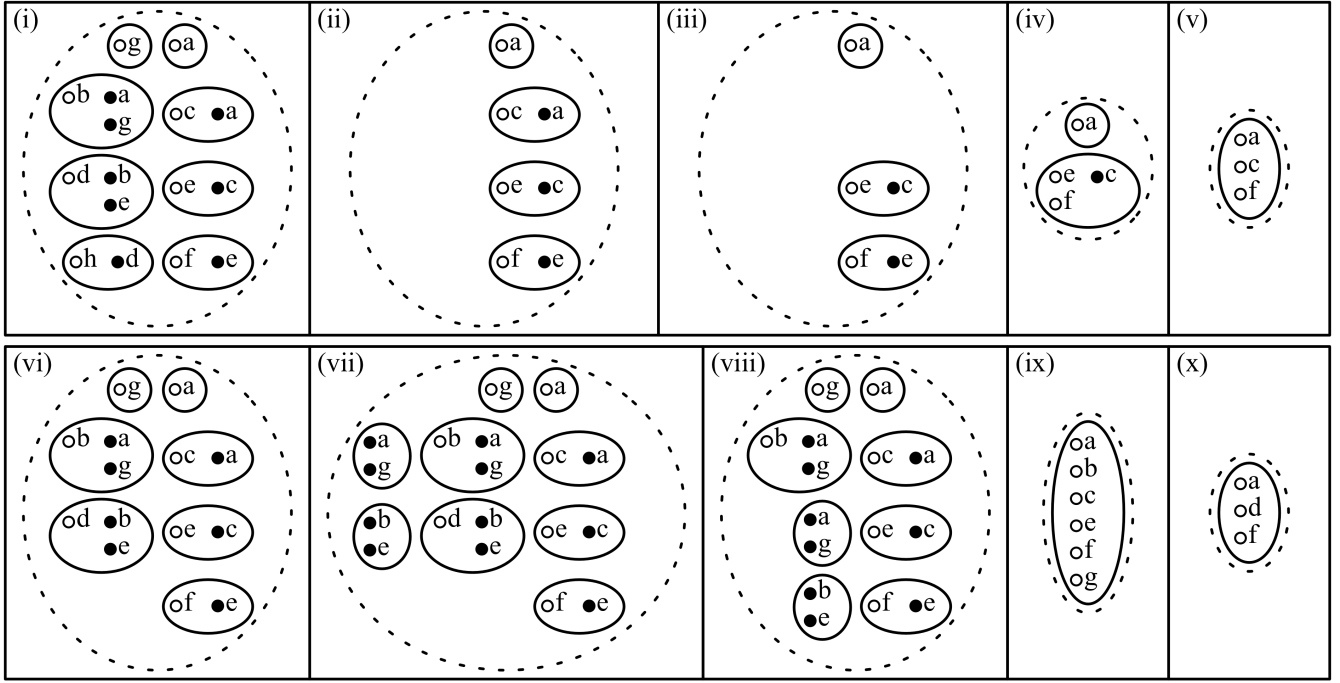


Figure 4: Testing adaptation twice in the DN  $\mathcal{D}$  in (i).

$p(d, be)$  is deleted as depicted in Figure 4 (viii). Recursively merging populations in step (iv) yields Figure 4 (ix). By definition,  $A(p(a), p(d, be), p(f, e))$  fails, since the population in Figure 4 (ix) contains  $a_c$  and  $f_c$ .

## Evolution

As promised, population classification is extended.

Let  $\mathcal{P}_Y = \{p(t_c, D) \mid p(t_c, D) \in \mathcal{D} \text{ and } t_d \in D'\}$  and  $\mathcal{P}_Z = \{p(t_c, D) \mid p(t_c, D) \in \mathcal{D} \text{ and } t_c \in C'\}$ , given DNs  $\mathcal{D}$  and  $D' = p(C', D')$ . In  $\mathcal{D}$ ,  $p(t_c, D)$  is *independent*, if  $A(p(t_c, D), \mathcal{P}_Y, \mathcal{P}_Z)$  succeeds, and is *evident*, if  $t_d$  is strong, and  $D$  is all relict. Population  $p(C, D)$  in a DN  $\mathcal{D}$  is *spent*, if there exists  $p(C', D)$  in  $\mathcal{D}$  such that  $C' \subset C$  and  $C - C'$  is all relict. In Figure 5, with  $\mathcal{D}$  in (ii) and  $D' = p(e, b)$  in (xiii),  $p(a)$  is independent as  $A(p(a), p(b, a), p(e, cd))$  succeeds, where  $\mathcal{P}_Y = p(b, a)$  and  $\mathcal{P}_Z = p(e, cd)$ . In  $\mathcal{D}$  of (iii) and  $D'$  of (xiii),  $p(b, a)$  is evident as  $b_d$  is strong, and  $a_d$  is relict. In  $\mathcal{D}$  of (vi),  $p(ce, dh)$  is spent as  $p(e, dh)$  is in  $\mathcal{D}$  and  $c_c$  is relict.

New populations can be created in a DN as follows. *Replication* of a population  $p(C, D)$  gives  $p(C, D)$ , as well as any set of populations  $p(C', D)$ , where  $C' \subset C$ .

The *evolution* of a DN  $\mathcal{D}$  into a DN  $\mathcal{D}'$  occurs by natural selection removing recursively all barren, independent, and spent populations, merging existing populations, and replicating to form new populations.

**Example 4.** In Figure 5, consider one explanation of the evolution of  $\mathcal{D}$  in (i) into  $\mathcal{D}' = p(e, b)$  in (xiii). Natural selection removes barren populations  $p(g, ef)$  and  $p(f, a)$ ,

yielding (ii). Next, natural selection removes independent population  $p(a)$ , giving (iii), and evident population  $p(b, a)$ , yielding (iv). Then,  $p(c, h)$  and  $p(e, cd)$  merge to form  $p(ce, dh)$  in (v). Replication gives (vi). The rest of the example involves natural selection (vii), merge (viii), replication (ix), natural selection (x), merge (xi), replication (xii), and natural selection (xiii), leaving  $\mathcal{D}'$  with population  $p(e, b)$ .

## Testing Independencies

Testing independencies in BNs can be seen as testing adaptation in DNs. D-separation can use specialized terminology not referenced in inference such as open sequential valves and closed divergent valves. In contrast, no specialized concepts are used in adaptation. And whereas m-separation requires DAGs, sub-DAGs, and moral graphs, adaptation uses but one platform.

$\mathcal{D} = \{p(v, Pa(v)) \mid P(v|Pa(v)) \text{ is in } \mathcal{B}\}$  is the DN for a given BN  $\mathcal{B}$ . Conversely, the *directed graph* (or simply *graph*)  $\mathcal{G}(\mathcal{D})$  of a DN  $\mathcal{D}$  has variables  $T_c(\mathcal{D})$  and arcs  $\{(v_i, v_j) \mid p(C, D) \in \mathcal{D} \text{ and } v_i \in D \text{ and } v_j \in C\}$ . The *undirected graph*  $U(\mathcal{D})$  of a DN  $\mathcal{D}$  has variables  $T_c(\mathcal{D})$  and edges  $\{(v_i, v_j) \mid p(C, D) \in \mathcal{D} \text{ and } v_i, v_j \in CD\}$ .

**Lemma 1.** Every BN  $\mathcal{B}$  can be represented as a DN  $\mathcal{D}$ , and the graph of  $\mathcal{D}$  is  $\mathcal{B}$ , that is,  $\mathcal{G}(\mathcal{D}) = \mathcal{B}$ .

The BN  $\mathcal{B}$  in Figure 1 (i) can be represented as the DN  $\mathcal{D}$  in Figure 4 (i). The graph of  $\mathcal{D}$  is  $\mathcal{B}$ , i.e.,  $\mathcal{G}(\mathcal{D}) = \mathcal{B}$ .

Let  $\mathcal{D}$  be the DN for a BN  $\mathcal{B}$  on  $U$ . The populations for  $W \subseteq U$ , denoted  $\mathcal{P}_W$ , are  $\mathcal{P}_W = \{p(C, D) \mid p(C, D) \in \mathcal{D} \text{ and } C \subseteq W\}$ . Thus, given pairwise disjoint subsets

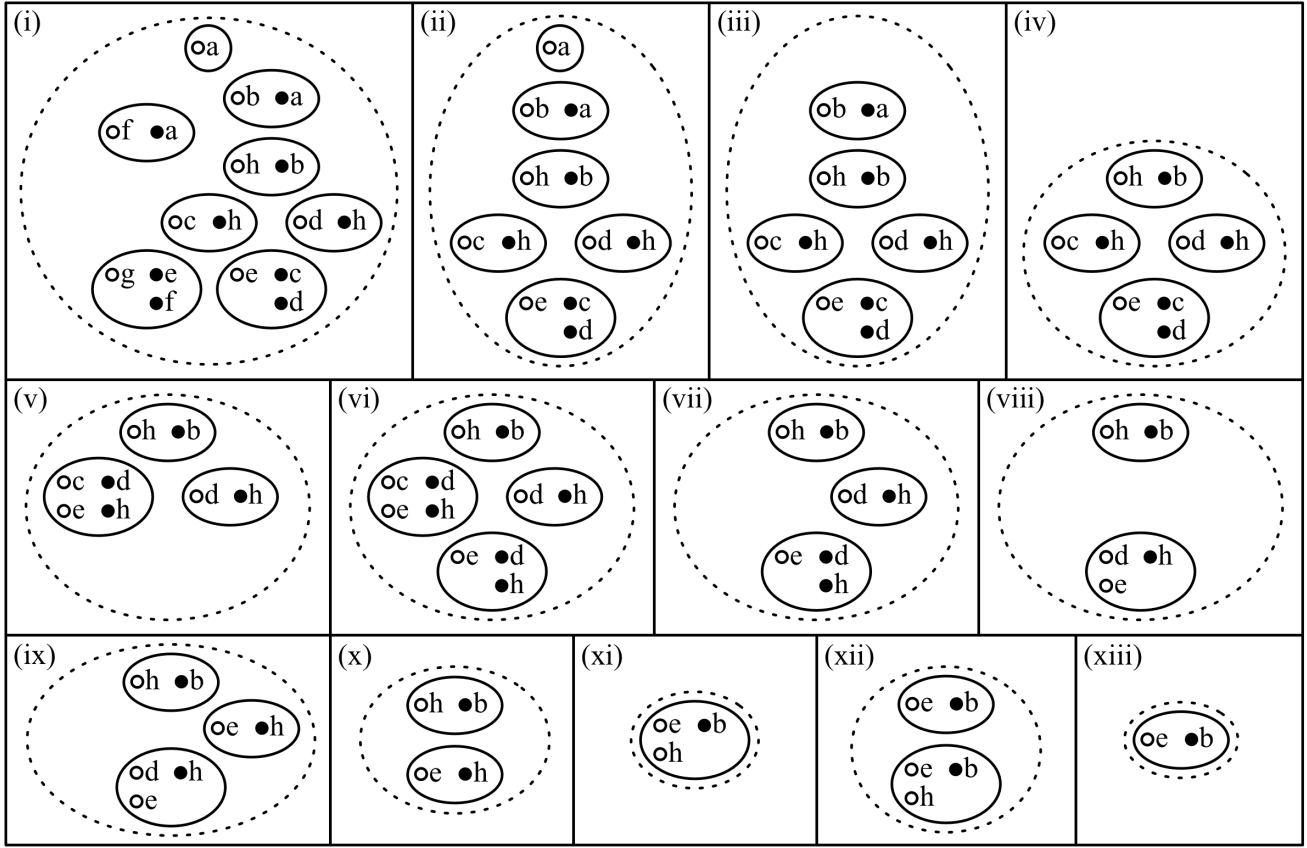


Figure 5: Representing VE's computation in Figure 2 as evolution in DNs.

$X, Y$ , and  $Z$  in  $\mathcal{B}$ , it is necessarily the case that  $\mathcal{P}_X, \mathcal{P}_Y$ , and  $\mathcal{P}_Z$  are pairwise disjoint populations in  $\mathcal{D}$ .

**Lemma 2.** Let  $\mathcal{B}^s$  be the sub-DAG constructed from a BN  $\mathcal{B}$  in step (i) of testing the independence  $I_{\mathcal{B}}(X, Y, Z)$  using m-separation. Then  $\mathcal{B}^s = \mathcal{G}(\mathcal{D}^s)$ , where  $\mathcal{D}^s$  is the DN constructed in step (i) of testing  $A(\mathcal{P}_X, \mathcal{P}_Y, \mathcal{P}_Z)$  in the DN  $\mathcal{D}$  for  $\mathcal{B}$ .

Step (i) of m-separation when testing  $I_{\mathcal{B}}(a, d, f)$  in the BN  $\mathcal{B}$  of Figure 1 (i) constructs the sub-DAG  $\mathcal{B}^s$  in Figure 1 (ii). On the other hand, step (i) of adaptation when testing  $A(p(a), p(d, be), p(f, e))$  in the DN  $\mathcal{D}$  in Figure 4 (i) constructs the DN  $\mathcal{D}^s$  in Figure 4 (vi). As guaranteed by Lemma 2,  $\mathcal{B}^s = \mathcal{G}(\mathcal{D}^s)$ .

**Lemma 3.**  $\mathcal{B}_m^s = U(\mathcal{D}_m^s)$ , where  $\mathcal{B}_m^s$  is the moralization of  $\mathcal{B}^s$  in Lemma 2, and  $\mathcal{D}_m^s$  is the docilization of  $\mathcal{D}^s$  in Lemma 2.

Recall the moralization  $\mathcal{B}_m^s$  in Figure 1 (iii) and the docilization  $\mathcal{D}_m^s$  in Figure 4 (vii), when testing  $I_{\mathcal{B}}(a, d, f)$  and  $A(p(a), p(d, be), p(f, e))$ , respectively. As Lemma 3 guarantees,  $\mathcal{B}_m^s = U(\mathcal{D}_m^s)$ .

**Lemma 4.** The undirected graph of the DN obtained by deleting the populations in  $\mathcal{P}_Y$  from  $\mathcal{D}_m^s$  is the same graph obtained by deleting  $Y$  and its incident edges from  $\mathcal{B}_m^s$ , where  $\mathcal{D}_m^s$  and  $\mathcal{B}_m^s$  are in Lemma 3.

When testing  $A(p(a), p(d, be), p(f, e))$ , deleting population  $p(d, be)$  in  $\mathcal{P}_Y$  from Figure 4 (vii) gives Figure 4 (viii). The undirected graph of the DN in Figure 4 (viii) is Figure 1 (iv). This is the same graph obtained by deleting variable  $d$  and incident edges from  $\mathcal{B}_m^s$  in Figure 1 (iii) in testing  $I_{\mathcal{B}}(a, d, f)$  using m-separation.

**Theorem 1.**  $I_{\mathcal{B}}(X, Y, Z)$  holds in a BN  $\mathcal{B}$  if and only if  $A(\mathcal{P}_X, \mathcal{P}_Y, \mathcal{P}_Z)$  succeeds in the DN  $\mathcal{D}$  for  $\mathcal{B}$ .

Theorem 1 indicates that testing adaptation in DNs can be used to test independencies in a BN  $\mathcal{B}$  replacing d-separation and m-separation.  $I_{\mathcal{B}}(a, c, f)$  holds by d-separation in Figure 1 (i) and  $A(p(a), p(c, a), p(f, e))$  succeeds in Example 2. Similarly,  $I_{\mathcal{B}}(a, d, f)$  does not hold in Figure 1 (i) by m-separation and  $A(p(a), p(d, be), p(f, e))$  fails as shown in Example 3.

The docilization step can be refined to add  $p(\emptyset, D)$  only for  $p(C, D) \in \mathcal{P}_Y$  with  $|D| > 1$ . Adding  $p(\emptyset, D)$  for  $p(C, D) \notin \mathcal{P}_Y$  is extraneous, since the merge of  $p(C, D)$  and  $p(\emptyset, D)$  is  $p(C, D)$ . Similarly, the moralization step in m-separation need only to add edges between parents of a common child  $v$  when  $v \in Y$ . For instance, in the moralization of Figure 1 (iii) when testing  $I_{\mathcal{B}}(a, d, f)$ , edge  $(b, e)$  is essential as  $d \in Y$ , but edge  $(a, g)$  is superfluous as  $b \notin Y$ .

## Performing Inference

Recall how VE computes query  $P(e|b = 0)$  posed to the BN  $\mathcal{B}$  in Figure 2 (i). Referring to Figure 5,  $\mathcal{B}$  is  $\mathcal{D}$  in (i), while  $P(e|b = 0)$  is DN  $\mathcal{D}'$  in (xiii). The removal of barren populations  $p(g, ef)$  and  $p(f, a)$  in (ii) corresponds to VE pruning barren variables  $g$  and  $f$  in Figure 2 (ii). Natural selection removes independent population  $p(a)$  in (iii) and VE removes independent by evidence variable  $a$  in Figure 2 (iii). VE then builds  $1(b)$  for the evidence variable  $b$ , while natural selection removes evident population  $p(b, a)$  in (iv). As for the elimination of  $c, d$ , and  $h$  in (1) - (4): the multiplication in (1) is the merge of  $p(c, h)$  and  $p(e, cd)$  in (iv), yielding  $p(ce, dh)$  in (v); the marginalization in (2) is the replication  $p(ce, dh)$  and  $p(e, dh)$  in (vi), followed by the removal of spent population  $p(ce, dh)$  in (vii); (3) is shown in (vii) - (x); and, (4) is in (x) - (xiii).

The robustness of DNs only is partially revealed in this example in which DNs detect and remove barren variables, detect and remove an independent by evidence variable, and represent multiplication and marginalization to eliminate variables  $c, d$ , and  $h$ . DNs can also represent AR (Butz, Oliveira, and dos Santos 2015a) and LP (Butz, Oliveira, and dos Santos 2015d). Next, we show how DNs can determine elimination orderings.

## Elimination Orderings

The order in which variables are eliminated can have profound impact on the amount of computation performed. Undirected graphs are typically used to determine good orderings (Koller and Friedman 2009).

**Example 5.** In Figure 6, consider the BN in (i) and its moralization in (ii). Eliminating variable  $t$  by adding edges between  $t$ 's neighbours and then removing  $t$  and its incident edges gives the undirected graph in Figure 6 (iii). Eliminating  $t$  from the BN yields the following factorization:

$$P(a) \cdot P(r|t) \cdot P(s) \cdots P(d|b, r). \quad (5)$$

Note that the undirected graph of (5) is exactly Figure 6 (iii). A more important point, unfortunately, is that the factorization corresponding to the undirected graph in Figure 6 (ii) is not necessarily unique. For example, the factorization

$$P(a|r, l), P(r|a, x), P(s|b) \cdots P(d). \quad (6)$$

defined by the very different BN in Figure 6 (iv) also gives the undirected graph in Figure Figure 6 (ii). Thus, given only the undirected graph to work with, it is not clear whether the corresponding probability tables are those in (5) or those in (6). DNs, on the contrary, maintain a one-to-one correspondence between the graphical representation and the probability tables (Butz, Oliveira, and dos Santos 2015b).

## Conclusion

DNs, a biological perspective of BNs, are surprisingly simple, yet remarkably robust. DNs can represent the testing of independencies using d-separation and m-separation, belief update using VE, AR, and LP. DNs simplify each of these separate techniques, while unifying them into one platform. Moreover, DNs can determine good elimination orderings.

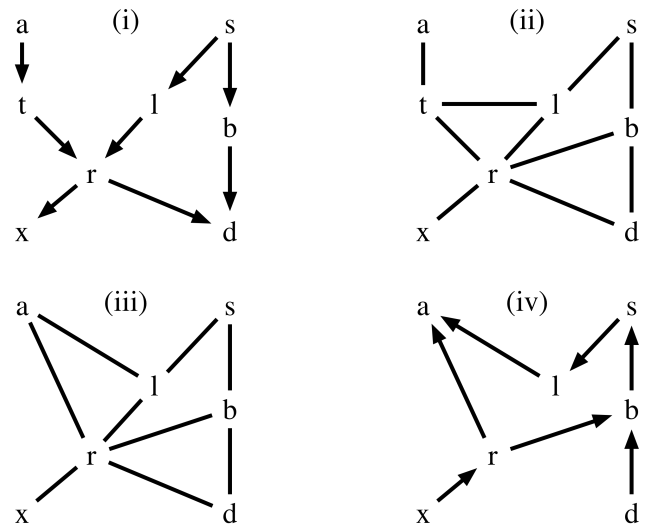


Figure 6: (i) a DAG. (ii) the moralization. (iii) eliminating  $t$ . (iv) another DAG giving the undirected graph in (iii).

## References

- Butz, C. J.; Oliveira, J. S.; and dos Santos, A. E. 2015a. Darwinian networks. In *Proceedings of the Twenty-Eighth Canadian Artificial Intelligence Conference (to appear)*.
- Butz, C. J.; Oliveira, J. S.; and dos Santos, A. E. 2015b. Determining good elimination orderings with darwinian networks. In *Proceedings of the Twenty-Eighth International FLAIRS Conference (to appear)*.
- Butz, C. J.; Oliveira, J. S.; and dos Santos, A. E. 2015c. On testing independencies in darwinian networks. Unpublished.
- Butz, C. J.; Oliveira, J. S.; and dos Santos, A. E. 2015d. Representing lazy propagation in darwinian networks. Unpublished.
- Darwiche, A. 2009. *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press.
- Dawkins, R. 1976. *The Selfish Gene*. Oxford U. Press.
- Koller, D., and Friedman, N. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.
- Lauritzen, S. L.; Dawid, A. P.; Larsen, B. N.; and Leimer, H. G. 1990. Independence properties of directed markov fields. *Networks* 20:491–505.
- Madsen, A. L., and Jensen, F. V. 1999. Lazy propagation: A junction tree inference algorithm based on lazy evaluation. *Artificial Intelligence* 113(1-2):203–245.
- Olmsted, S. 1983. *On Representing and Solving Decision Problems*. Ph.D. Dissertation, Stanford University.
- Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.
- Zhang, N. L., and Poole, D. 1994. A simple approach to bayesian network computations. In *Proceedings of the Tenth Canadian Artificial Intelligence Conference*, 171–178.