Semantics of Multiply Sectioned Bayesian Networks for Cooperative Multi-agent Distributed Interpretation

Y. Xiang

Department of Computer Science, University of Regina
Regina, Sask., Canada S4S 0A2, yxiang@cs.uregina.ca

Abstract. In order to represent cooperative multi-agents who must reason with uncertain knowledge, a coherent framework is necessary. We choose multiply sectioned Bayesian networks (MSBNs) as the basis for this study because they are based on well established theory on Bayesian networks and because they are modular. In this paper, we focus on the semantics of a MSBN-based multi-agent system (MAS) for cooperative distributed interpretation. In particular, we establish the conditions under which the joint probability distribution of a MSBN-based MAS can be meaningfully interpreted. These conditions imply that a coherent MSBN-based MAS can be constructed using agents built by different developers. We show how the conditions can be satisfied technically under such a context. (*Keywords*: Knowledge representation, probabilistic reasoning, multi-agent systems.)

1 Introduction

Bayesian networks (BNs) [10, 9, 7, 6] provide a coherent formalism for representing and reasoning with uncertain knowledge in AI systems. As commonly applied, a BN assumes a single-agent paradigm. That is, a single processor accesses a single global network representation, updates the joint probability distribution (jpd) over the domain variables as evidence becomes available and answers queries. A multiply sectioned Bayesian networks (MSBN) [20] is a set of interrelated Bayesian subnets over a large problem domain decomposed into a set of loosely coupled subdomains. Each subnet encodes the knowledge about a subdomain. MSBNs allow modular knowledge representation in large domains and facilitate efficient inference computation [19].

As originally developed, a MSBN is intended for a single-agent system and as an aid to probabilistic inference of a single user. In such a system, the MSBN encodes the knowledge of a single developer/expert. The jpd represents the coherent belief of the expert. This is the *semantics* of a single-agent MSBN.

An agent in a multi-agent system (MAS) [1, 2, 4, 14] is an autonomous intelligent subsystem. Each agent holds its partial domain knowledge, accesses an external information source and consumes some computational resource. Each communicates with others in achieving its goal. Agents may be *cooperative* in achieving a common goal or may be *self-interested* with conflicting goals. Agents may be *homogeneous* or *heterogeneous*. Concurrent approaches in MASs are essentially logic-based, which do not have a coherent framework for representing agents with uncertain knowledge. The fundamental question that inspired this study is "In order to represent cooperative multi-agents who must reason with uncertain knowledge, what would be a proper framework?" We choose MSBNs as a basis for this study because they are based on well established theory on BNs and therefore are coherent and general (no built-in ad-hoc assumptions), and because they are modular,

We consider the extension of MSBNs into *cooperative* and *homogeneous* MASs.[1] In particular, the extension is intended for *distributed interpretation* tasks. As defined by Lesser and Erman [8], an *interpretation* system accepts evidence from some environment and produces higher level descriptions of objects and events in the environment. A *distributed* interpretation system is needed when sensors for collecting evidence are distributed, and communication of all evidence to a centralized site is undesirable. Potential applications include sensor networks, diagnosis and trouble-shooting of complex systems, distributed image interpretation, etc.

Figure 1 shows major components of an agent in a MSBN-based MAS. The subnet is the central component that holds the knowledge and belief of the agent on the subdomain. The reasoner is responsible to update the belief when evidence is obtained from local sensors. The communicator is responsible to perform belief propagation among agents. The sensitivity analyzer suggests the most valuable evidence to acquire next based on the current belief. The decision maker determines the actions that affect the external world. The structure verifier verifies the correctness of global structure through distributed operations.

To extend single-agent MSBNs into MASs, many issues need to be resolved. Earlier works involve the coherent agent communication [15], the optimization of communication scheduling [16], and the distributed structure verification [18]. The focus of this paper is the semantics of a MSBN-based MAS:

- What is the interpretation of the jpd of the MSBN and under what conditions such an interpretation is well-defined?
- How can we build a coherent MSBN-based MAS by multiple developers?
- What is the advantage of a MSBN-based MAS over a set of BN-based agents without organized as a MSBN?

The rest of the paper is organized as follows. Section 2 briefly introduces single-agent MSBNs. Section 3 establishes the semantics of MSBN-based MASs. Section 4 discusses technical issues in constructing a coherent MSBN by multiple developers. Section 5 analyzes why agents should be organized into a MSBN for probabilistic inference. Section 6 presents an example.

[1] It is our belief that unless we understand well how to perform uncertain inference coherently in a homogeneous MAS, we are less likely to succeed in dealing with the issue in a heterogeneous MAS.

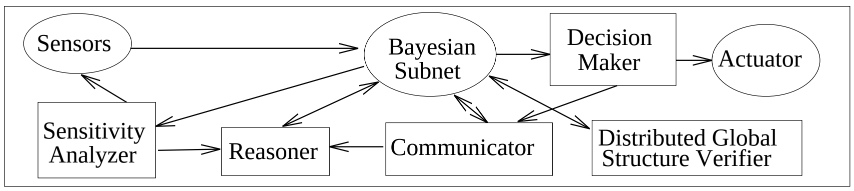

Fig. 1. Major components of an agent in a MSBN-based MAS.

2 Single-Agent MSBNs

To make the paper self-contained, we briefly introduce the single-agent MSBNs [20, 19]. A BN [10, 9, 7, 6] is a triplet $S = (N, E, P)$. N is a set of nodes. Each node is labeled with a variable associated with a space. We shall use 'node' and 'variable' interchangeably. Hence N represents a problem domain. E is a set of arcs such that $D = (N, E)$ is a directed acyclic graph (DAG). We shall refer to D as the *structure* of the BN. The arcs signify directed dependencies between the linked variables. For each node $A_i \in N$, the strengths of the dependencies on the set of parent nodes π_i are quantified by a conditional probability distribution $p(A_i|\pi_i)$. For any three sets X, Y and Z of variables, X and Y are said to be *conditionally independent* given Z under probability distribution P if $P(X|YZ) = P(X|Z)$ whenever $P(YZ) > 0$. The basic dependency assumption embedded in BNs is that a variable is conditionally independent of its non-descendants given its parents. This allows the jpd P to be specified by the product $P = \prod_i p(A_i|\pi_i)$.

A MSBN M consists of a set of interrelated Bayesian subnets over a large problem domain or *total universe*. Each subnet represents dependencies of a subdomain and shares a non-empty set of variables with at least one other subnet. The intersection between each pair of subnets satisfies a *d-sepset* condition.

Definition 1 (d-sepset) *Let $D^i = (N^i, E^i)$ $(i = 1, 2)$ be two DAGs such that $D = (N^1 \cup N^2, E^1 \cup E^2)$ is a DAG. The intersection $I = N^1 \cap N^2$ is a* `d-sepset` *between D^1 and D^2 if, for every $A_i \in I$ with its parents π_i in D, either $\pi_i \subseteq N^1$ or $\pi_i \subseteq N^2$. A node in a d-sepset is called a* `d-sepnode`*.*

The condition essentially requires that for each node in the d-sepset, at least one subnet contains all its parent nodes. It can be shown that, when a pair of subnets are isolated from M, their d-sepset renders them conditionally independent. Figure 2 (left) shows the structure of a *trivial* MSBN for diagnosis of Median nerve lesion (Medn), Carpal tunnel syndrome (Cts) and Plexus upper trunk lesion (Plut). It consists of three subnets D^i $(i = 1, 2, 3)$ for clinical, electromyography (EMG) and nerve conduction subdomains, respectively. The d-sepset between each pair of subnets is $\{Medn, Cts, Plut\}$. [2]

Subnets of M are organized into a *hypertree* structure. Each hypernode is a subnet of M. Each hyperlink is a d-sepset between a pair of subnets. A hypertree is so structured that it ensures that each hyperlink render the two parts of M that it connects conditionally independent. The subnets in Figure 2 (left) can be organized into the hypertree in Figure 2 (middle). Figure 2 (right) depicts a general hypertree structured MSBN (unrelated to the one in the middle).

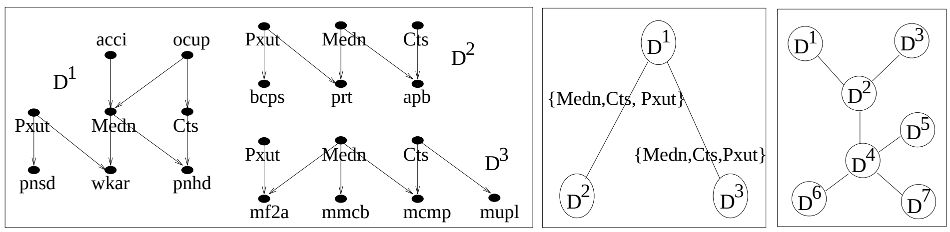

Fig. 2. Left: An example MSBN for neural muscular diagnosis. Middle: The hypertree organization of the MSBN in the left. Right: A general hypertree structured MSBN.

Each subnet in M may be multiply connected (more than one path between a pair of nodes), e.g., D^1. In order to perform inference more efficiently in each subnet, the hypertree structured M is converted into a *linked junction forest* (LJF) F of the identical structure as its run time representation. Each hypernode in the hypertree is a *junction tree* (JT) (clique tree) converted from the corresponding subnet through moralization and triangulation [7, 6]. Each hyperlink in the hypertree is a set of *linkages* which covers the d-sepset between the two corresponding subnets.

The need for linkages can be understood as follows: When evidence is obtained in one subnet/JT, it can be propagated to an adjacent JT by passing the probability distribution over the d-sepset I. This may not be efficient if the cardinality of I is large. The efficiency can be improved by exploiting the conditional independence within I. Linkages form a decomposition of I based on conditional independence. Once linkages are defined, the probability distribution over I can be passed by passing distributions over linkages, which is more efficient [17]. Linkages are obtained as follows:

Definition 2 (linkage) *Let I be the d-sepset between JTs T^a and T^b in a LJF.*

First remove recursively every leaf clique C of T^a that satisfies one of the following conditions. (1) $C \cap I = \phi$. (2) $C \cap I$ is a subset of another clique. Denote the resultant graph by T'.

[2] In general, d-sepsets between different pairs of subnets of M may be different.

Then remove recursively either a member variable from a clique of T' or a clique from T' as follows. (a) If a variable $x \notin I$ is contained in a single clique C, remove x from C. (b) If a clique C becomes a subset of an adjacent clique D after (a), union C into D.

The resultant is a `linkage tree` *$Y^{a \to b}$ of T^a relative to T^b. Each clique l of $Y^{a \to b}$ is a* `linkage` *from T^a to T^b.*

It can be shown that a linkage tree is a JT. It can also be shown that belief propagation between JTs through linkages can be performed coherently if and only if $Y^{a \to b}$ and $Y^{b \to a}$ are identical.

The MSBN in Figure 2 (left and middle) can be converted into the LJF in Figure 3. The three subnets D^i $(i = 1, 2, 3)$ are converted into three JTs T^i $(i = 1, 2, 3)$. Then linkages (shown as heavy links) between pairs of JTs are defined. The linkage tree of T^2 relative to T^1 is obtained by first removing the clique $C8$, and then removing the variable *apb* from the clique $C6$ and removing *prt* from $C7$. We then obtained the linkage tree with two cliques $\{Cts, Medn\}$ and $\{Pxut, Medn\}$ each of which is a linkage between T^1 and T^2.

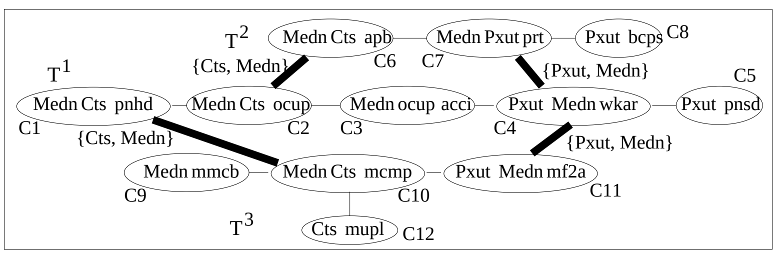

Fig. 3. A linked junction forest of the MSBN in Figure 2.

Parallel to the structural conversion, the conditional probability tables stored at nodes of M are converted to *belief tables* (unnormalized probability distributions) of cliques in JTs of F such that a *joint system belief* of F, assembled from the belief tables, is equivalent to the jpd of M. The belief table of a JT T is

$$B_T(N) = \prod_i B_{C_i}(C_i) / \prod_j B_{S_j}(S_j), \tag{1}$$

where N is the set of domain variables of T, $B_{C_i}(C_i)$ is the belief table of clique C_i and $B_{S_j}(S_j)$ is the belief table of clique separator S_j. Subscripts are used to denote the object that a belief table is associated with. Let $B_I(I)$ be the belief table of a d-sepset I assembled from belief tables of linkages in the corresponding linkage tree in the similar fashion as Equation 1 (recall that a linkage tree is a JT). The joint system belief of F takes the form

$$B_F(U) = \prod_i B_{T^i}(N^i) / \prod_j B_{I^j}(I^j), \tag{2}$$

where $U = \cup_i N^i$ is the total universe. Since belief tables are unnormalized probability distributions, $B_F(U)$ is proportional to the jpd of F

$$P_F(U) = \prod_i P_{T^i}(N^i) / \prod_j P_{I^j}(I^j), \tag{3}$$

where P denotes a probability distribution.

To answer queries by local computation in F, it must be consistent. F is *locally consistent* if all JTs are internally consistent, i.e., when marginalized onto the same set of variables, different belief tables in a JT yield the identical marginal distribution. F is *boundary consistent* if each pair of adjacent JTs are consistent with respect to their d-sepset. F is *globally consistent* if it is both locally consistent and boundary consistent. A set of operations are defined to achieve consistency during evidential reasoning.

Suppose F is initially globally consistent. Details on initialization can be found in the above reference. After evidence is entered into a JT, the JT is no longer internally consistent and F is no longer globally consistent. Evidence is entered by the operation `EnterEvidence`. `EnterEvidence` multiplies the belief tables of relevant cliques with the evidence function and then brings the JT internally consistent by an outward belief propagation and then an inward belief propagation within the JT [6].

For example, suppose Median motor conduction block ($mmcb = true$) and Median to Ulnar palmar latency difference ($mupl > 0.7ms$) are observed in the nerve conduction study of a patient. In Figure 3, the clique $C9$ contains the variable *mmcb* and the clique $C12$ contains the variable *mupl*. During `EnterEvidence`, first the belief tables of $C9$ and $C12$ will be modified such that belief of all configurations incompatible with the observation will be set to 0. Then a clique is arbitrarily selected, say, $C9$. Afterwards, belief propagates inwards from $C11$ to $C10$ and from $C12$ to $C10$, and then belief propagates from $C10$ to $C9$. After the inward propagation, belief propagates outwards from $C9$ to $C10$, and then from $C10$ to $C11$ and $C12$. This brings T^3 internally consistent.

For belief propagation operations to maintain global consistency in single-agent MSBNs, readers are referred to the above reference. We review the communication operations for maintaining global consistency in a MSBN-based MAS in Section 5.

3 The Interpretation of Jpd in a MSBN-based MAS

As described in Section 2, a MSBN represents a large problem domain by representing each subdomain with a subnet. From the viewpoint of a reasoning agent, a MSBN represents the *coherent* multiple perspectives of a single agent. For example, PAINULIM [19] consists of three subnets which represents a neurologist's three different perspectives of the neuromuscular diagnostic domain: clinical, EMG and nerve conduction perspectives. The jpd of the MSBN represents the subjective belief of a single expert.

In a MSBN-based MAS, each agent can be considered as holding its partial perspective of the domain. The modularity of MSBN allows its natural extension to a MAS: Instead of representing *one* agent's *multiple* perspectives of a domain, a MSBN-based MAS represents *multiple* agents in a domain each of which holds *one* distinct perspective of the domain. Each subnet corresponds to one such perspective. A natural question then arises: What is the interpretation of the jpd of such a system? Whose belief does it represent? We will first discuss this issue intuitively and then justify our interpretation formally.

Consider a computer system. It processes information coherently as a whole, even though its components are commonly supplied by different developers. This coherence is achieved since each developer follows a set of protocols in designing the functional *interface* of a component. As long as the interface follows a common protocol, a developer has the freedom to determine the internal structure of a component and the entire system will function as if it follows a single mind. How much knowledge is necessary to the integrator of the system? He only needs to know the functional interfaces of components and not their internal structures. In a sense, the system is built by a *group* of designers including all developers who supply components as well as the system integrator. Building complex systems in such a way has become a common practice. Procedural abstraction and data abstraction are commonly applied to develop complex software systems by team work [5]. Layered approach is commonly used in operating systems [13] and computer networks [12].

Next consider a human 'system' consisting of a patient and a family doctor. Suppose that the patient has *no* medical knowledge of her problem and she trusts the doctor's expertise *completely*. Suppose that the doctor is also giving the best diagnosis and treatment he can. When they meet, the patient tells all that the doctor needs to know for diagnosis. After the doctor reaches a diagnosis, he prescribes a therapy which the patient follows. Even though the doctor does not experience the symptom himself and the patient does not understand how the diagnosis is reached, the system as a whole demonstrates a coherent belief on symptoms (the doctor uses to reach the diagnosis) and the diagnosis (the patient follows the therapy). Situations like this are not uncommon when a user is seeking advice from a specialist. Who is the integrator of this system? It's the demand and supply (of medical expertise).

The two scenarios illustrate that, under certain conditions, a system consisting of different agents may demonstrate a joint belief coherent with that of each individual agent. Clearly one of the conditions is that agents are *cooperative*, also termed *benevolence* [14]. An agent must trust the information supplied by others and must also supply others with what he really believes, which is termed *veracity* [14]. This is possible if all agents in the system are working towards a common goal (vs self-interested).

Another condition is *conditional independence*. It is not necessary for each agent to supply others with *all* that he believes. A component in a complex system only needs to pass to other components the information specified in the protocol, and it can and should hide other details regarding how the supplied information is obtained. In structured programming, a procedure header only specifies the input and output parameters. How the mapping from input to output is performed needs not be concerned by the caller of the procedure. A doctor only needs to inform the patient of the diagnosis and the therapy. He does not need to explain how the diagnosis is reached. In general, to a particular agent engaged in a particular task, there is usually a certain amount of information from other agents, once exchanged, that is sufficient to help the agent to perform its own task. Beyond that amount, the information about how other agents think is irrelevant, namely, the agent is conditionally independent of other agents conditioned on that certain amount of information.

We now formalize the above idea by applying a result from statistics [3] which was not intended for MASs. We first introduce a third condition.

Definition 3 *Let N be a set of variables in a problem domain. Let A and B be two subsets of N such that $A \cap B \neq \phi$ and $A \cup B = N$. Let $Q(A)$ and $R(B)$ be probability distributions over A and B. $Q(A)$ and $R(B)$ are said to be* `consistent` *if $\sum_{A \setminus B} Q(A) = \sum_{B \setminus A} R(B)$, where the summation represents marginalization.*

In other words, $Q(A)$ and $R(B)$ are consistent if they yield the same distribution when marginalized to $A \cap B$. The following lemma is due to Dawid and Lauritzen and is reformulated in our notation.

Lemma 4 *[3] Let $N = A \cup B$ be a set of variables. Let $Q(A)$ and $R(B)$ be probability distributions over A and B and let them be consistent. Then there exists a* unique *probability distribution*

$$P(N) = Q(A)R(B|A \cap B) \text{ whenever } R(A \cap B) > 0$$

such that (1) $\sum_{N \setminus A} P(N) = Q(A)$, (2) $\sum_{N \setminus B} P(N) = R(B)$ and (3) A is conditionally independent of B given $A \cap B$ under P.

Now let α and β be two cooperative agents. Suppose α can only perceive the subdomain A and β can only perceive the subdomain B. Let the subjective belief of α be represented by $Q(A)$ and that of β be represented by $R(B)$. Suppose knowing the other agent's belief on the intersection $A \cap B$ is sufficient to coordinate the tasks of α and β and $Q(A)$ and $R(B)$ are consistent. Then, according to Lemma 4, there exists a unique probability distribution $P(N)$ that is identical to $Q(A)$ when restricted to A and identical to $R(B)$ when restricted to B, and that it satisfies the conditional independence of A and B conditioned on $A \cap B$.

Relating the above discussion to MSBN-based MASs, we can represent agents α and β by a MSBN M with two subnets S^α and S^β over subdomains A and B such that their d-sepset is $A \cap B$. The distribution of S^α corresponds to α's belief and the distribution of S^β corresponds to β's belief. The boundary consistency of F (the LJF of M) corresponds to the consistency of the two agents' belief. When F is globally consistent, the jpd defined by Equation 3 is identical to $P(N)$ in Lemma 4. The following theorem generalizes Lemma 4 to the case of more than two distributions.

Theorem 5 *[3] Let N be a set of variables. Let T be a junction tree and C_i be a clique of T such that $\cup_i C_i = N$. Let $Q_{C_i}(C_i)$ be the probability distribution over the clique C_i such that distributions for each pair of adjacent cliques in T are consistent. Let S_j be a clique separator in T and $Q_{S_j}(S_j)$ be the distribution over S_j computed from the distribution of any one of its adjacent cliques. Then there exists a* unique *probability distribution*

$$P_T(N) = \prod_i Q_{C_i}(C_i) / \prod_j Q_{S_j}(S_j)$$

such that (1) for each clique C_i, $\sum_{N \setminus C_i} P_T(N) = Q_{C_i}(C_i)$ and (2) for each pair of adjacent cliques C_i and C_j with their separator S_k, C_i is conditionally independent of C_j given S_k under P_T.

According to Theorem 5, if we organize a set of *cooperative* agents into a MSBN such that adjacent agents in the hypertree are *conditionally independent* and *consistent*, then the jpd of the MSBN defines a coherent joint belief among all agents. This joint belief is identical to the belief of each agent when restricted to the corresponding subdomain and it supplements each agent's limited knowledge outside the agent's subdomain by the knowledge of other agents. Since the requirements conditional independence and consistency are only restrictions on the *interface* (vs internal structure) of agents, a MSBN-based MAS can be constructed using agents built by multiple developers. Each developer builds one computational agent (a subnet) based on its own expertise in a subdomain.

4 How to Make Adjacent Agents Consistent?

As shown in Section 3, adjacent subnets in the hypertree structure of a MSBN should be *consistent* as defined in Definition 3. We address the technical issues for ensuring this consistency.

When no confusion arises, we refer to the *structure of a subnet* as simply the *subnet*. If a d-sepnode x has no parent in the entire MSBN, we call x a *global root*. Clearly, a global root is a root in every subnet that contains it. For example, $Pxut$ in Figure 2 is a global root. According to Definition 1, if a d-sepnode x is not a global root, then for each pair of parents y and z of x, whenever y is contained in a subnet, z must be contained in that subnet as well. Hence for a d-sepnode x that is not a global root and for each subnet D that contains x, either x is a root in D or x and all its parents are contained in D. For example, Cts in Figure 2 is a root in D^2 but a non-root in D^1.

By Definition 3, adjacent subnets in a MSBN are consistent if their distribution on their d-sepset are identical. Since the distribution on a d-sepset is determined by the distributions on d-sepnodes, in order to ensure consistency of subnets, we must ensure distributions on d-sepnodes are identical across subnets. When agents are built by a single developer, this imposes no problem. We discuss how to ensure the consistency when agents are built by different developers.

For a d-sepnode x that is a global root, each subnet that contains x must associate x with a prior distribution, which may be assigned differently by different developers. A brute force method would adopt the prior from one of the developers. A more natural method can be devised using the idea in [11] for combining probabilities from multiple experts. We illustrate the method as follows: Suppose x has k possible outcomes $x \in \{x_1, x_2, \ldots, x_k\}$. Instead of letting each developer specify a prior distribution $(p_1, p_2, \ldots, p_k)$ for x for the corresponding subnet, $k{+}1$ non-negative integers $(n_1, n_2, \ldots, n_k, m)$ are supplied by each developer such that $\sum_{i=1}^{k} n_i = m$ and $p_i = n_i/m$ $(i = 1, \ldots, k)$. The ratio n_i/m is interpreted as though the developer had observed x_i for n_i times out of m trials. Now the prior for x in each subnet can be assigned as $(\sum_j n_{1j}/\sum_j m_j, \ldots, \sum_j n_{kj}/\sum_j m_j)$, where the index j is over each subnet containing x.

For a d-sepnode y that is not a global root, if y appears as a non-root in a subnet D, a probability distribution of y conditioned on its parents is associated with y in D. If y appears as a root in D', a prior distribution is associated with x in D'. Suppose y appears as a non-root in j subnets and as a root in k subnets. The conditional distribution of y in the first j subnets can be determined in the same way as we combine the prior distributions of a global root illustrated above. Once this is determined, the prior distribution of y in the k subnets where y appears as a root are constrained by the assignment of the conditional distribution in the first j subnets as well as other relevant distributions in the entire MSBN. The actual numerical parameters can be determined through belief propagation during the initialization process mentioned in Section 3. Details on initialization can be found in [20].

5 MSBNs Ensure Disciplined Communication

One might wonder what difference a MSBN-based MAS makes compared to the same set of agents without being organized into a MSBN. Can each agent cooperate with others by sending messages and treating messages received as evidence? We first review briefly the operation `CommunicateBelief` for maintaining global consistency in a MSBN-based MAS [15] and then answer this question.

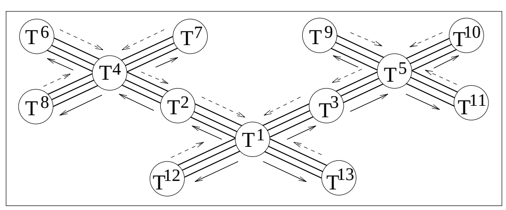

Fig. 4. Belief propagation during `CommunicateBelief` in a LJF. Each node represents a JT. Multiple links between two nodes represents multiple linkages between the JTs. The operation is initiated from T^1.

Without going into the formal details, Figure 4 shows how belief propagates through a LJF during `CommunicateBelief`. Each node is a JT and corresponds to an agent in the system. Multiple links between two adjacent JTs corresponds to multiple linkages between them. Suppose the operation is initiated at an arbitrarily selected JT T^1. First control is propagated from T^1 towards terminal JTs along solid arrows, and then the belief tables on d-sepsets are propagated from terminal JTs back to T^1 along dotted arrows. Afterwards, belief is propagated from T^1 towards terminal JTs along solid arrows. Note that since a LJF is not equivalent to a JT due to the existence of multiple linkages, belief propagation in a LJF is not the same as in a JT representing a single BN [6]. The issue not dealt with by the algorithm for inference in a JT is that propagation must be performed through multiple linkages coherently. See [20, 17] for details. It can be shown that after `CommunicateBelief`, the LJF is globally consistent. That is, the answers to queries from each agent are coherent with all evidence gathered in the entire system [15].

It should now be clear that the belief propagation in MSBNs during `CommunicateBelief` is in fact message passing. The messages are the belief tables over linkages. However, message passing in a MSBN is *disciplined*.

First, messages in a MSBN must flow along the hypertree in a regulated fashion as illustrated above. Now suppose a set of agents are not organized as a MSBN (or some equivalent form). Then the following sequence of events is possible. Initially an agent α may send a message to an agent β based on a piece of evidence. After updating its belief based on the message and some additional local evidence, β may send a message to an agent γ. After updating its belief based on the message and some additional local evidence, γ may send a message to α. Not knowing the message from γ is based *partially* on the evidence originated from α, α will update its belief and count the same evidence twice. Such circular evidence propagation causes *no* problem if the knowledge of all agents is *deterministic* or *logical*. However, it will create *false* belief with no evidential support if the knowledge of agents is *uncertain* or *probabilistic* [10]. The hypertree structure of MSBNs and the way `CommunicateBelief` operates ensures that no circular evidence propagation occur among agents.

Furthermore, the hypertree structure of a MSBN is not just any tree structure, just as a clique tree of a BN cannot be just any tree but should be a JT. Recall from Section 2, the hypertree is so organized such that each d-sepset renders the two parts of the hypertree that it connects *conditionally independent*. A detailed description is beyond the scope of this paper and can be found in [20]. The point is that a MSBN requires that the belief on the entire d-sepset between a pair of subnets, in the form of belief tables over all linkages, be passed each time. This information, passed in the hypertree with the above property and in the way in which `CommunicateBelief` is defined, is sufficient to ensure a coherent joint system belief. On the other hand, if agents are organized in an arbitrary tree structure, even though circular evidence propagation can be avoided, it still cannot ensure a coherent joint system belief.

6 Illustration of Multi-Agent MSBNs

Let us consider monitoring or troubleshooting a complex artifact system as shown in Figure 5. The system is made of five subsystems $U0$ through $U4$. The set of external input variables of each subsystem is labeled by I, each set of data

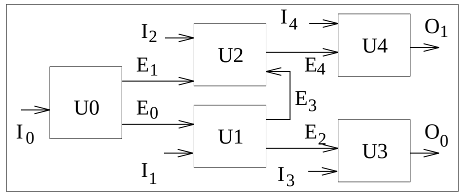

Fig. 5. An artifact consisting of five subsystems.

flowing from one subsystem to another is labeled by E, and each set of output from a subsystem to the external world is labeled by O. Suppose subsystems are manufactured by different developers. Each developer also builds an agent (whose central component is a Bayesian subnet) that encodes the knowledge of the functional and the faulty behavior of parts, and of the internal structure of the subsystem. Each agent is capable of monitoring or troubleshooting the corresponding subsystem. To monitor the entire artifact system, we can construct a MAS and let those agents cooperate.

We assume that external inputs are independent of each other and there is no feedback between subsystems as is the case in this example. Then each agent is independent of others given the variables that connect the agent to others. The distributions for those variables can be set using the techniques discussed in Section 4. Now all the semantic conditions required (cooperative, conditional independent and consistent) are met and we can organize the agents into a MSBN-based MAS.

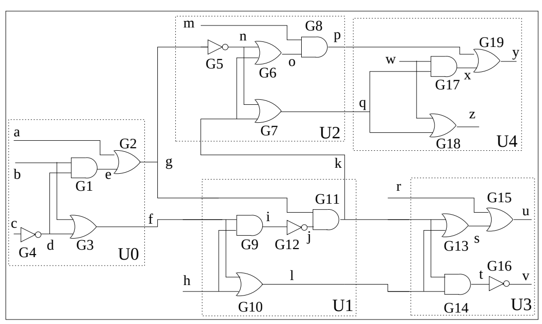

Fig. 6. A digital system consisting of five circuits.

The above illustration is independent of the particular application domain of an artifact system. To make the example more concrete, we fill each box of Figure 5 with a digital circuit as in Figure 6.

Note that, in integrating the MSBN, only the knowledge of the interface of each circuit as shown in Figure 5 is needed. The knowledge of the internal structure of each circuit is not necessary. Furthermore, although the function of each gate is commonly defined, its faulty behavior may vary from subsystem to subsystem. For example, $U1$ and $U2$ may be supplied by different developers. An AND gate in $U1$ may have the stuck-at-0 faulty behavior, but an AND gate in $U2$ may output correctly 40% of time when it is faulty. The developer of each circuit, not the integrator of the entire system, is in the best position to encode such knowledge, and such knowledge can be hidden (by not disclosing the circuit configuration and the structure and the distributions of the subnet) from the integrator if so desired. If a circuit from a developer is replaced by another with the same functional interface but from a different developer, we simply replace the corresponding subnet (effectively replacing the corresponding agent) without disturbing the rest of the MSBN. The new MSBN-based MAS will still perform coherently.

Figure 7 (left) shows the five Bayesian subnets for the five circuits in Figure 6. Figure 7 (right) shows the hypertree organization of the MSBN. Such a MAS may be used to aid a group of users each interacting with one computational agent by entering local evidence and querying.

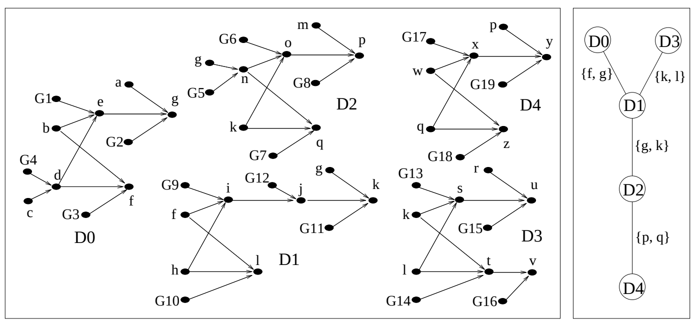

Fig. 7. Left: The Bayesian subnets for the five circuits in Figure 6. Right: The hypertree organization of the five subnets as a MSBN.

7 Remarks

In this paper, we have shown that for cooperative agents to perform probabilistic inference coherently in a distributed interpretation task, they can be organized into a MSBN-based MAS or some equivalent structure. We established that if agents are cooperative, conditionally independent and consistent, then the jpd of the MSBN is identical to each agent's belief when restricted to the agent's subdomain, and is supplementary to the agent's limited knowledge outside the agent's subdomain.

The latter two conditions (conditional independence and consistency) are constraints to only interfaces between agents. Therefore, a coherent MSBN-based MAS can be intergrated from agents built by different developers and it is not necessary for developers to disclose the internal structures of their subnets.

We indicate that the development of MSBNs are motivated by coherent inference with uncertain knowledge in large problem domains, and therefore all example MSBNs given in the paper are *trivial* and are only used for the purpose of illustration of the framework. To use either single-agent or multi-agent MSBNs in practice, the problem domain should be decomposable into loosely coupled subdomains. This requirement dictates the size of individual subdomains. Each subdomain should not be too large since it defeats the purpose of using a MSBN. Each subdomain should not be too small either since subdomains will then be densely coupled and it again defeats the purpose of MSBNs.

Acknowledgement

This work is supported by Research Grant OGP0155425 from NSERC. Helpful comments from anonymous reviewers are acknowledged.

References

1. A.H. Bond and L. Gasser. An analysis of problems and research in dai. In A.H. Bond and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 3–35. Morgan Kaufmann, 1988.
2. A.H. Bond and L. Gasser, editors. *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann, 1988.
3. A.P. Dawid and S.L. Lauritzen. Hyper Markov laws in the statistical analysis of decomposable graphical models. *Annals of Statistics*, 21(3):1272–1317, 1993.
4. L. Gasser and M.N. Huhns, editors. *Distributed Artificial Intelligence, Volume II*. Morgan Kaufmann, 1989.
5. C. Ghezzi, M. Jazayeri, and D. Mandrioli. *Fundamentals of Software Engineering*. Prentice Hall, 1991.
6. F.V. Jensen, S.L. Lauritzen, and K.G. Olesen. Bayesian updating in causal probabilistic networks by local computations. *Computational Statistics Quarterly*, (4):269–282, 1990.
7. S.L. Lauritzen and D.J. Spiegelhalter. Local computation with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B*, (50):157–244, 1988.
8. V.R. Lesser and L.D. Erman. Distributed interpretation: a model and experiment. *IEEE Trans. on Computers*, C-29(12):1144–1163, 1980.
9. R.E. Neapolitan. *Probabilistic Reasoning in Expert Systems*. John Wiley and Sons, 1990.
10. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
11. D. Poole, A. Mackworth, and R. Goebel. *Computational Intelligence: A Logical Approach*. Oxford University Press, forthcoming, 1996.
12. W.A. Shay. *Understanding Data Communications and Networks*. PWS Publishing, 1995.
13. A. Silberschatz and P.B. Galvin. *Operating System Concepts*. Addison Wesley, 1994.
14. M. Wooldridge and N.R. Jennings. Intelligent agents: theory and practice. *Knowledge Engineering Review*, 10(2):115–152, 1995.
15. Y. Xiang. Distributed multi-agent probabilistic reasoning with Bayesian networks. In Z.W. Ras and M. Zemankova, editors, *Methodologies for Intelligent Systems*, pages 285–294. Springer-Verlag, 1994.
16. Y. Xiang. Distributed scheduling of multiagent systems. In *Proc. 1st Inter. Conf. on Multi-agent Systems*, pages 390–397, San Francisco, 1995.

17. Y. Xiang. Optimization of inter-subnet belief updating in multiply sectioned Bayesian networks. In *Proc. 11th Conf. on Uncertainty in Artificial Intelligence*, pages 565–573, Montreal, 1995.
18. Y. Xiang. Distributed structure verification in multiply sectioned Bayesian networks. In *Proc. 9th Florida Artificial Intelligence Research Symposium*, pages 295–299, Key West, 1996.
19. Y. Xiang, B. Pant, A. Eisen, M. P. Beddoes, and D. Poole. Multiply sectioned Bayesian networks for neuromuscular diagnosis. *Artificial Intelligence in Medicine*, 5:293–314, 1993.
20. Y. Xiang, D. Poole, and M. P. Beddoes. Multiply sectioned Bayesian networks and junction forests for large knowledge based systems. *Computational Intelligence*, 9(2):171–220, 1993.

This article was processed using the LaTeX macro package with LLNCS style