# On the Role of Multiply Sectioned Bayesian Networks to Cooperative Multiagent Systems

Y. Xiang
University of Guelph, Canada, yxiang@cis.uoguelph.ca
V. Lesser
University of Massachusetts at Amherst, USA, lesser@cs.umass.edu

***Abstract*—We consider a common task in multiagent systems where agents need to estimate the state of an uncertain domain so that they can act accordingly. If each agent only has partial knowledge about the domain and local observations, how can the agents accomplish the task with a limited amount of communication? Multiply sectioned Bayesian networks (MSBNs) provide an effective and exact framework for such a task but also impose a set of constraints. Are there simpler frameworks with the same performance but with less constraints? We identify a small set of high level choices which logically imply the key representational choices leading to MSBNs. The result addresses the necessity of constraints of the framework. It facilitates comparisons with related frameworks and provides guidance to potential extensions of the framework. (Keywords: multiagent system, decentralized interpretation, communication, organization structure, uncertain reasoning, probabilistic reasoning, belief network, Bayesian network)**

## I. Introduction

As intelligent systems are being applied to larger, open and more complex problem domains, many applications are found to be more suitably addressed by multiagent systems [25], [27]. Consider a large uncertain problem domain populated by a set of agents. The agents are often charged with many tasks determined by the nature of the application. One common task is to estimate what is the true state of the domain so that they can act accordingly. Such a task, often referred to as *distributed interpretation* [15], arises in many applications of multiagent systems including equipment/process troubleshooting, building/area surveillance, battle field/disaster situation assessment, and distributed design. We can describe the domain with a set of variables. Some variables are not directly observable hence their values can only be inferred based on observations of other variables and background knowledge about their dependence relations. Furthermore, each agent has only a partial perspective of the problem domain. That is, each agent only has knowledge about a subdomain, i.e., about the dependence among a subset of domain variables, and can only observe and reason within the subdomain. The agents may be developed by different designers and the subdomain knowledge may be private to their designers. Hence, maintaining the privacy of the agents while they are cooperating may be desirable.

In the case of a single agent, the task of estimating the state of the domain can be achieved by representing the domain knowledge in a Bayesian network (BN) [20] and by performing probabilistic inference using the BN given the agent's observations. However, as multiple agents are cooperating on the task, a set of new issues arise: How should the domain be partitioned into subdomains? How should each agent represent its knowledge about a subdomain? How should the knowledge of each agent relate to that of others? How should the agents be organized in their activities? What information should they exchange and how, in order to accomplish their task with a limited amount of communication? Can they achieve the same level of accuracy in estimating the state of the domain as that of a single centralized agent?

Multiply sectioned Bayesian networks (MSBNs) [28] provide one solution to these issues. An MSBN consists of a set of interrelated Bayesian subnets each of which encodes an agent's knowledge concerning a subdomain. Agents are organized into a hypertree structure such that inference can be performed in a distributed fashion while answers to queries are exact with respect to probability theory. Each agent only exchanges information with adjacent agents on the hypertree, and each pair of adjacent agents only exchange their beliefs on a set of shared variables. Both local inference within an agent and communication among all agents are efficient when the agent subnets are sparse. Therefore, MSBNs provide a framework in which multiple agents can estimate the state of a domain effectively[1] with exact and distributed probabilistic

[1] We shall use the term "effective" to mean *efficient* computation when agent subnets are sparse.

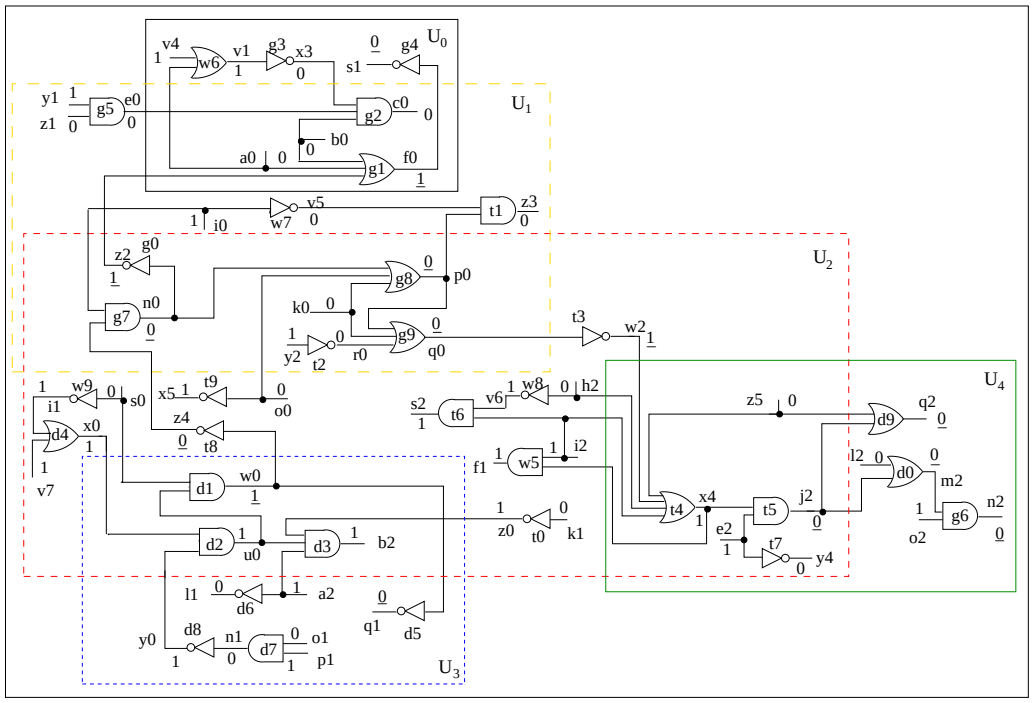



Fig. 1. A digital system.

inference. In principle, the framework allows unbounded number of agents as well as allows agents to join and leave dynamically.

Are there simpler alternatives that can achieve the same performance? In other words, are the technical constraints of MSBN necessary? For example, the hypertree organization of agents prevents an agent from communicating directly and arbitrarily with another agent. Is this necessary? The agent interface is required to satisfy a condition called *d-sepset* (detailed in the paper). Is it necessary?

In this work, we address these issues. We show that given some reasonable fundamental choice/assumptions, the key constraints of an MSBN, such as a hypertree structure and a d-sepset agent interface, follow logically. In particular, we identify the choice points in the formation of the MSBN framework. We term the fundamental choices as *basic commitments* (BCs). Given the BCs, other technical choices are entailed. Hence, an MSBN or some equivalent follows once we admit the BCs.

The contributions of this work are the following: First, the analysis provides a high-level (vs. technical level) description of the applicability of MSBN and addresses issues regarding the necessity of major MSBN representational constraints. Second, the results facilitate comparison with alternative frameworks. Third, the analysis provides a guideline for extensions or relaxations of the MSBN framework as to what can or cannot be traded off.

In Section II, we briefly overview the MSBN framework with representational choices summarized. Each remaining section identifies some BCs and derives implied choices.

## II. Overview of MSBNs

A BN [20] $S$ is a triplet $(V, G, P)$ where $V$ is a set of domain variables, $G$ is a DAG whose nodes are labeled by elements of $V$, and $P$ is a joint probability distribution (jpd) over $V$, specified in terms of a distribution for each variable $x \in V$ conditioned on the parents $\pi(x)$ of $x$ in $G$. An MSBN [33], [28] $M$ is a collection of Bayesian subnets that together define a BN. For instance, suppose that a piece of equipment consists of multiple components built by different designers. As a small example, Figure 1 shows a piece of digital equipment made out of five components $U_i$ $(i = 0, ..., 4)$. Each box in the figure corresponds to a component and contains the logical gates and their connections with the input/output signals of each gate labeled. A set of five agents, $A_i$ $(i = 0, ..., 4)$ respectively associated with a component $U_i$, cooperate to monitor the system and trouble-shoot it when necessary. Each agent $A_i$ is responsible for a particular component $U_i$, which is likely being developed by the designer of the component. When a gate is enclosed in exactly one box, the gate is *physically* located in the corresponding component and is *logically* known only to the agent responsible for the component. On the other hand, when a gate is enclosed in more than one box, the gate is *physically* located in only one of the components but is *logically* known to all the corresponding agents. For

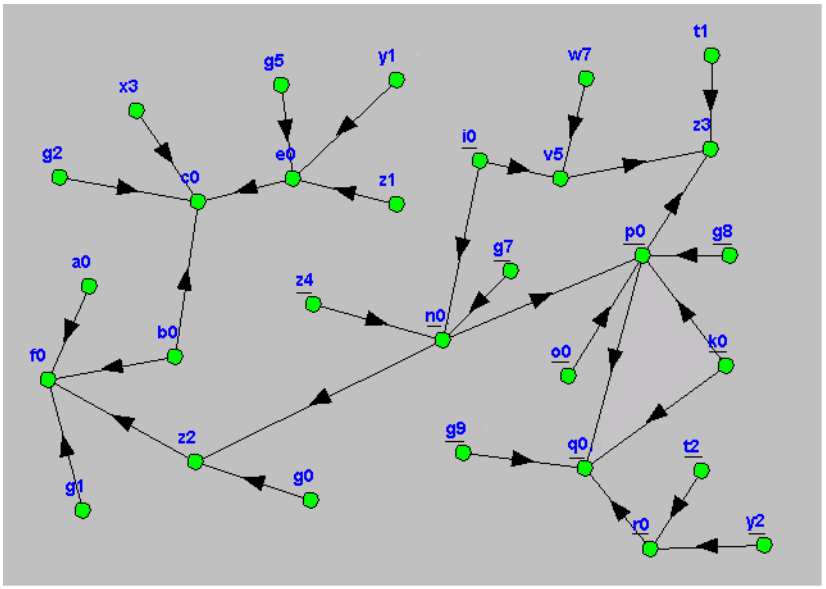



Fig. 2. The subnet $G_1$ for $U_1$.

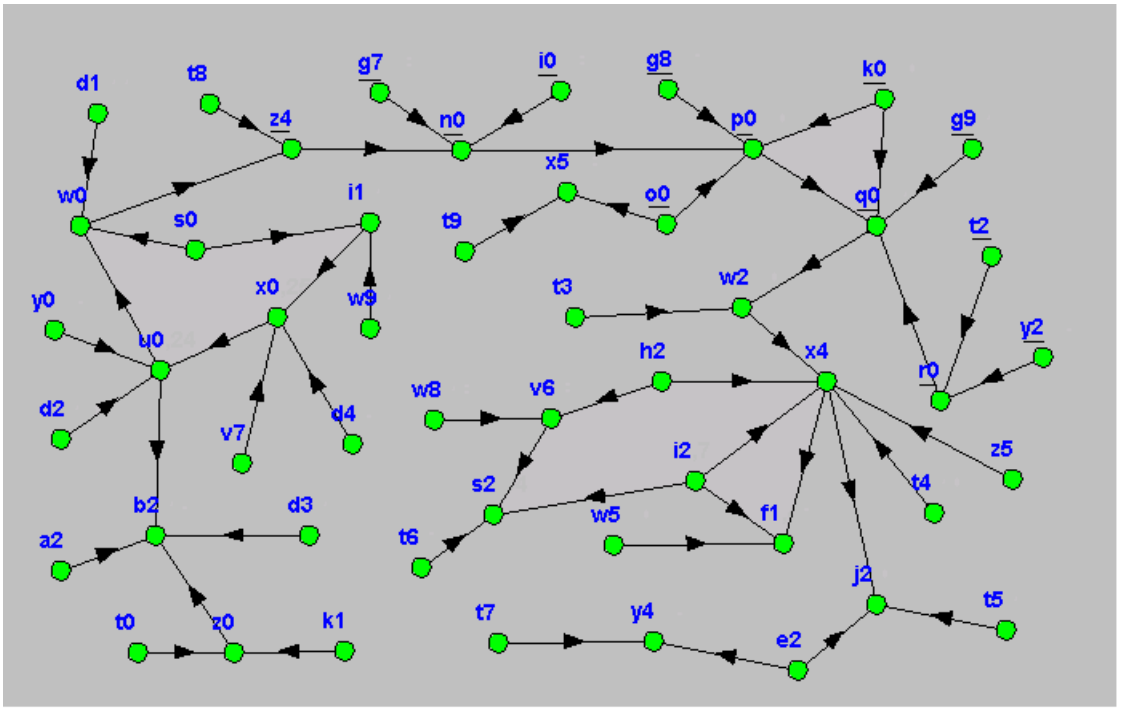



Fig. 3. The subnet $G_2$ for $U_2$.

example, the AND gate $g_5$ is known only to $A_1$, the OR gate $g_8$ is known to both $A_1$ and $A_2$, and the signal $z_2$ is known to $A_0$, $A_1$ and $A_2$. The knowledge of an agent about its assigned component can be represented as a BN, called a *subnet*. The subnet for agent $A_1$ (responsible for component $U_1$) is shown in Figure 2 and that for $A_2$ is shown in Figure 3. Each node is labeled with a variable name. Only the DAGs of the subnets are shown in the figures with the conditional probability distribution for each variable omitted. As mentioned above, an agent not only knows all devices located in its assigned component, but also knows some devices that are physically located in other interfacing components. Hence, each subnet encodes the agent's knowledge on both types of devices. The five subnets (one for each component) collectively define an MSBN, which form the core knowledge of the multi-agent system. Based on this knowledge and limited observations, agents can cooperate to estimate whether the system is functioning normally, and if not, which devices are likely to be responsible. For instance, suppose that the gates $d_1$ (in $U_3$) and $t_5$ (in $U_4$) in Figure 1 break down and produce incorrect output. Some outputs downstream are also affected. Equipment inputs and correct device outputs are shown in Figure 1 by 0 and 1. Incor-

rect outputs are shown underlined. Through limited local observation (each agent observes the values of 3 to 4 signals incrementally) and communication (two rounds), agents can identify the two faulty gates [31], [28] correctly (with $P(d_1 = faulty|all\ observations) = 0.98$ and $P(t_5 = faulty|all\ observations) = 0.99$).

Subnets in an MSBN are required to satisfy certain conditions. To describe these conditions, we introduce the terminologies first. Let $G_i = (V_i, E_i)$ $(i = 0, 1)$ be two graphs (directed or undirected). $G_0$ and $G_1$ are said to be *graph-consistent* if the subgraphs of $G_0$ and $G_1$ spanned by $V_0 \cap V_1$ are identical. Given two graph-consistent graphs $G_i = (V_i, E_i)$ $(i = 0, 1)$, the graph $G = (V_0 \cup V_1, E_0 \cup E_1)$ is called the *union* of $G_0$ and $G_1$, denoted by $G = G_0 \cup G_1$. Given a graph $G = (V, E)$, a partition of $V$ into $V_0$ and $V_1$ such that $V_0 \cup V_1 = V$ and $V_0 \cap V_1 \neq \emptyset$, and subgraphs $G_i$ of $G$ spanned by $V_i$ $(i = 0, 1)$, $G$ is said to be *sectioned* into $G_0$ and $G_1$. See Figure 4 for an example. Note that if $G_0$ and $G_1$ are

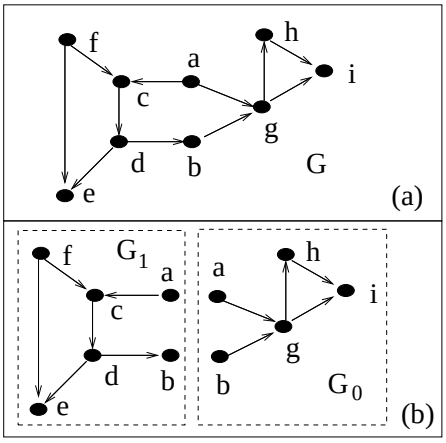



Fig. 4. The graph $G$ in (a) is sectioned into $G_0$ and $G_1$ in (b). $G$ is the union of $G_0$ and $G_1$.

sectioned from a third graph, then $G_0$ and $G_1$ are graph-consistent. The union of multiple graphs and the sectioning of a graph into multiple graphs can be similarly defined.

Graph sectioning is useful in defining the dependence relation between variables shared by agents. It is used to specify the following hypertree condition which must be satisfied by subnets in an MSBN:

*Definition 1:* Let $G = (V, E)$ be a connected graph sectioned into subgraphs $\{G_i = (V_i, E_i)\}$. Let the subgraphs be organized into an undirected tree $\Psi$ where each node is uniquely labeled by a $G_i$ and each link between $G_k$ and $G_m$ is labeled by the non-empty `interface` $V_k \cap V_m$ such that for each $i$ and $j$, $V_i \cap V_j$ is contained in each subgraph on the path between $G_i$ and $G_j$ in $\Psi$. Then $\Psi$ is a `hypertree` over $G$. Each $G_i$ is a `hypernode` and each interface is a `hyperlink`.

Figure 5 illustrates a hypertree for the digital system, where $G_1$ and $G_2$ are shown in Figures 2 and 3. The

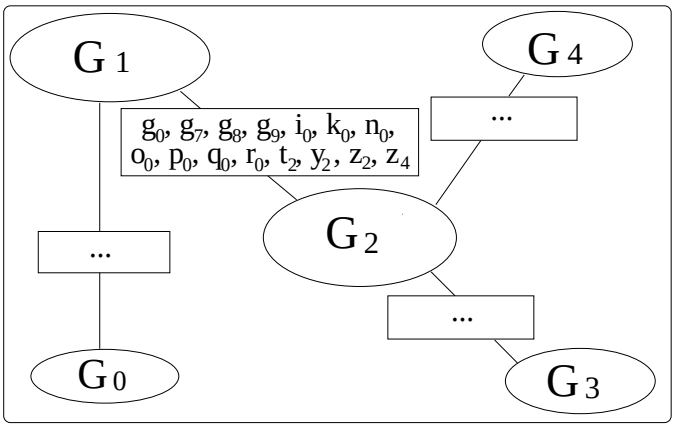



Fig. 5. The hypertree for the digital equipment monitoring system.

hypertree represents an organization of agent communication, where variables in each hypernode are local to an agent and variables in each hyperlink are shared by agents. Agents communicate in an MSBN by exchanging their beliefs over shared variables. We use *nodes* and variables interchangeably when there is no confusion. Nodes shared by subnets in an MSBN must form a *d-sepset*, as defined below:

*Definition 2:* [2] Let $G$ be a directed graph such that a hypertree over $G$ exists. A node $x$ contained in more than one subgraph with its parents $\pi(x)$ in $G$ is a `d-sepnode` if there exists at least one subgraph that contains $\pi(x)$. An interface $I$ is a `d-sepset` if every $x \in I$ is a d-sepnode.

The interface between $G_1$ and $G_2$ contains 13 variables indicated in Figure 5. The corresponding nodes in Figures 2 and 3 are underlined. It is a d-sepset because these variables are only shared by $G_1$ and $G_2$, and each variable has all its parents contained in one of them. For instance, the parents of $z_4$ ($t_8$ and $w_0$) are all contained in $G_2$, while those of $n_0$ ($i_0$, $g_7$ and $z_4$) are contained in both $G_1$ and $G_2$ (see Figures 2 and 3). The structure of an MSBN is a multiply sectioned DAG (MSDAG) with a hypertree organization:

*Definition 3:* A `hypertree MSDAG` $G = \bigcup_i G_i$, where each $G_i$ is a DAG, is a connected DAG such that (1) there exists a hypertree $\Psi$ over $G$, and (2) each hyperlink in $\Psi$ is a d-sepset.

Note that although DAGs in a hypertree MSDAG form a tree, each DAG may be multiply connected. A *loop* in a graph is a sequence of nodes $a, b, c, ..., a$ such that the first node is identical to the last node and there is a link

[2] Note that this definition is an extension of earlier definitions for d-sepset, such as that in [28], to the most general case.

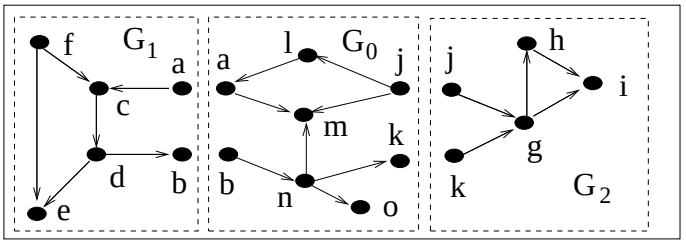



Fig. 6. A MSDAG with multiple paths across local DAGs.

(not necessarily in the same direction) between each pair of nodes adjacent in the sequence. Such a loop is also referred to as an *undirected loop*. A DAG is *multiply connected* if it contains at least one (undirected) loop. Otherwise, it is singly connected. For example, $G_1$ in Figure 2 has two loops. One of them is $(i_0, v_5, z_3, p_0, n_0, i_0)$. Hence, $G_1$ is multiply connected. $G_2$ in Figure 3 has several loops and is also multiply connected. Moreover, multiple paths may exist from a node in one DAG to another node in a different DAG after the DAGs are unioned. For instance, in Figure 6, there are several (undirected) paths from node $c$ in $G_1$ to node $g$ in $G_2$. There is one path going through nodes $a$, $l$ and $j$ and another path goes through $d$, $b$, $n$ and $k$. Each path goes across all three DAGs.

An agent's quantitative knowledge about the strength of dependence of a variable $x$ on its parent variables $\pi(x)$ can be encoded as a conditional probability distribution $P(x|\pi(x))$. $P(x|\pi(x))$ is a special case of a *potential* over $\{x\} \cup \pi(x)$. A potential over a set of variables is an non-negative distribution of at least one positive parameter. For instance, the following table illustrates a potential $B(x, y)$ over two variables $x$ and its parent $y$. It

TABLE I

A POTENTIAL $B(x, y)$ OVER A SET $\{x, y\}$ OF VARIABLES THAT REPRESENTS A PROBABILITY DISTRIBUTION $P(x|y)$

| $x$ | $y$ | $B(x, y)$ | $P(x\|y)$ |
|---|---|---|---|
| 0 | 0 | 1 | 0.25 |
| 1 | 0 | 3 | 0.75 |
| 0 | 1 | 8 | 0.8 |
| 1 | 1 | 2 | 0.2 |

may represent the probability distribution $P(x|y)$ in the last column. One can always convert $B(x, y)$ into $P(x|y)$ by dividing each potential value with a proper sum: an operation termed *normalization*. For instance, after dividing the first two values in $B(x, y)$ by their sum 4, we obtain the probability values $P(x = 0|y = 0) = 0.25$ and $P(x = 1|y = 0) = 0.75$. After dividing the last two values in $B(x, y)$ by their sum 10, we obtain the probability values $P(x = 0|y = 1) = 0.8$ and $P(x = 1|y = 1) = 0.2$. Hence, $B(x, y)$ contains the same information as $P(x|y)$ with the flexibility of not having to perform the normalization until it is needed. A *uniform potential* is one with all its potential values being 1.

An MSBN is then defined as follows. Uniform potentials are used to ensure that quantitative knowledge about the strength of dependence of a variable on its parent variables will not be doubly specified for the same variable.

*Definition 4:* An MSBN $M$ is a triplet $(V, G, \mathcal{P})$. $V = \bigcup_i V_i$ is the `domain` where each $V_i$ is a set of variables. $G = \bigcup_i G_i$ (a hypertree MSDAG) is the `structure` where nodes of each DAG $G_i$ are labeled by elements of $V_i$. Let $x$ be a variable and $\pi(x)$ be all the parents of $x$ in $G$. For each $x$, exactly one of its occurrences (in a $G_i$ containing $\{x\} \cup \pi(x)$) is assigned $P(x|\pi(x))$, and each occurrence in other DAGs is assigned a uniform potential. $\mathcal{P} = \prod_i P_i$ is the `jpd`, where each $P_i$ is the product of the potentials associated with nodes in $G_i$. A triplet $S_i = (V_i, G_i, P_i)$ is called a `subnet` of $M$. Two subnets $S_i$ and $S_j$ are said to be adjacent if $G_i$ and $G_j$ are adjacent on the hypertree MSDAG.

MSBNs provide a framework for the task of estimating the state of an uncertain domain in cooperative multiagent systems. Each agent holds its partial perspective (a subnet) of a domain, reasons about the state of its subdomain with local observations and through limited communication with other agents. Each agent may be developed by an independent designer and the internals of an agent (agent privacy) are protected. Agents can acquire observations in parallel while their beliefs about the states of individual subdomains are consistent with observations acquired by all agents. For the digital system example, each component $U_i$ is assigned an agent $A_i$ in charge of the subnet $S_i$ and its local computation.

The representational choices of MSBNs are summarized below, where the most important ones are 3 and 6.

1) Each agent's belief is represented by Bayesian probability.
2) The domain is decomposed into subdomains. For each pair, there exists a sequence of subdomains such that every pair of subdomains adjacent in the sequence shares some variables.
3) Subdomains are organized into a (hyper)tree structure where each hypernode is a subdomain, and each hyperlink represents an non-empty set of shared variables between the two hypernodes such that variables shared by any two hypernodes are also shared by each hypernode on the path between

them.

4) The dependency structure of each subdomain is represented by a DAG.
5) The union of DAGs for all subdomains is a connected DAG.
6) Each hyperlink is a d-sepset.
7) The joint probability distribution can be expressed as in Definition 4.

Below we identify a set of BCs leading to these choices.

## III. On communication graphs

We use *uncertain knowledge*, *belief* and *uncertainty* interchangeably, and make the following basic commitment:

*BC 1:* Each agent's belief is represented by Bayesian probability.

It directly corresponds to the choice 1 of Section II. We shall use *coherence* to describe any assignment of belief consistent with the probability theory.

We consider a domain $V$ of variables populated by $n$ cooperative agents $A_0, ..., A_{n-1}$. Each $A_i$ has knowledge over $V_i \subset V$, called the *subdomain* of $A_i$. For example, in equipment monitoring, each $V_i$ corresponds to a component including all its devices and their input/output signals. Although not required in theory, practically it is assumed whenever $V_i \cap V_j \neq \emptyset$, the intersection is *small* relative to $V_i$ and $V_j$. From BC 1, the knowledge of $A_i$ is a probability distribution over $V_i$, denoted by $P_i(V_i)$.

To minimize communication, we allow agents to exchange only their beliefs on shared variables (BC 2 below). We take it for granted that for agents to communicate directly, $V_i \cap V_j$ must be nonempty. Note that BC 2 does not restrict the *order* nor the *number* of communications.

*BC 2:* $A_i$ and $A_j$ can communicate *directly* only with $P(V_i \cap V_j)$.

We refer to $P(V_i \cap V_j)$ as a *message* and refer to direct communication as *message passing*. We emphasize that the fundamental property of message passing (as used in this paper) is that the messages normally reveal only partial information known to the sender. In other words, neither a single message nor all the messages from a sender collectively disclose all the information that the sender has. Paths for message passing can be represented by a *communication graph* (CG): In a graph with $n$ nodes, associate each node with an agent $A_i$ and label it by $V_i$. Connect each pair of nodes $V_i$ and $V_j$ by a link labeled by $I = V_i \cap V_j$ if $I \neq \emptyset$. Figure 7 shows the communication graph of the multiagent system for monitoring the digital system from Figure 1. The subdomains and their intersections are shown below:

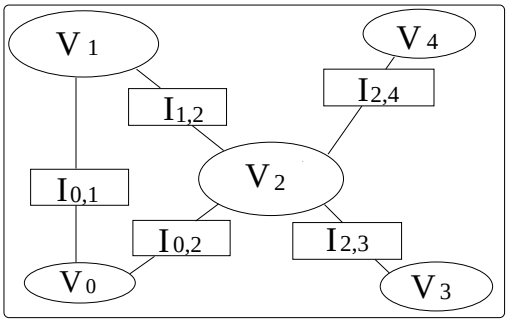



Fig. 7. The communication graph of the multiagent system for monitoring the digital system.

$$
\begin{aligned}
V_0 &= \{a_0, b_0, c_0, e_0, f_0, g_1, g_2, g_3, g_4, s_1, v_1, v_4, w_6, x_3, z_2\}\\
V_1 &= \{a_0, b_0, c_0, e_0, f_0, g_0, g_1, g_2, g_5, g_7, g_8, g_9, i_0, k_0, n_0, o_0, p_0, q_0, r_0, t_1, t_2, v_5, w_7, x_3, y_1, y_2, z_1, z_2, z_3, z_4\}\\
V_2 &= \{a_2, b_2, d_1, d_2, d_3, d_4, e_2, f_1, g_0, g_7, g_8, g_9, h_2, i_0, i_1, i_2, j_2, k_0, k_1, n_0, o_0, p_0, q_0, r_0, s_0, s_2, t_0, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, u_0, v_6, v_7, w_0, w_2, w_5, w_8, w_9, x_0, x_4, x_5, y_0, y_2, y_4, z_0, z_2, z_4, z_5\}\\
V_3 &= \{a_2, b_2, d_1, d_2, d_3, d_5, d_6, d_7, d_8, l_1, n_1, o_1, p_1, q_1, s_0, u_0, w_0, x_0, y_0, z_0\}\\
V_4 &= \{d_0, d_9, e_2, g_6, h_2, i_2, j_2, l_2, m_2, n_2, o_2, q_2, t_4, t_5, t_7, w_2, x_4, y_4, z_5\}
\end{aligned}
$$

$$
\begin{aligned}
I_{0,1} &= \{a_0, b_0, c_0, e_0, f_0, g_1, g_2, x_3, z_2\}\\
I_{0,2} &= \{z_2\}\\
I_{1,2} &= \{g_7, g_8, g_9, i_0, k_0, n_0, o_0, p_0, q_0, r_0, t_2, y_2, z_4\}\\
I_{2,3} &= \{a_2, b_2, d_1, d_2, d_3, s_0, u_0, w_0, x_0, y_0, z_0\}\\
I_{2,4} &= \{e_2, h_2, i_2, j_2, t_4, t_5, t_7, w_2, x_4, y_4, z_5\}
\end{aligned}
$$

In fact, a communication graph is an application of a general class of graphs called *junction graphs* [9]. Although our focus is on communication graphs, many of their relevant properties are intrinsic to all junction graphs. Therefore, we shall describe these properties in terms of junction graphs whenever it is appropriate. Definition 5 defines junction graphs formally. We use $2^V$ to denote the power set of a set $V$.

*Definition 5:* A `junction graph` is a triplet $(V, \Omega, E)$. $V$ is an non-empty set called the `generating set`. $\Omega$ is a subset of $2^V$ such that $\cup_{Q\in\Omega} = V$. Each element $Q$ of $\Omega$ is called a `cluster`. $E$ is defined as

$$E = \{\langle Q_1, Q_2\rangle | Q_1, Q_2 \in \Omega, Q_1 \neq Q_2, Q_1 \cap Q_2 \neq \emptyset\},$$

where each unordered pair $\langle Q_1, Q_2 \rangle$ is called a `separator` between the two clusters $Q_1$ and $Q_2$, and is labeled by the intersection $Q_1 \cap Q_2$.

In Figure 7, the generating set $V$ is the set of all variables in the digital system domain. Each cluster corresponds to one component subdomain. The separators in a CG represent all potential paths for message passing among agents. As the belief of one agent can influence the belief of another agent through a third agent, CG also represents all potential paths for *indirect* communications. Each agent's belief should potentially be influential in any other, directly or indirectly. Otherwise the system can be split into two. Hence, CG is *connected.* We summarize this in Proposition 6. It is equivalent to Choice 2 in Section II. Recall that BC stands for basic commitment.

*Proposition 6:* Let $H$ be the communication graph over $V$ that observes BC 1 and BC 2. If each agent's belief can in general influence that of each other agent through communication, then $H$ is connected.

A CG contains all possible paths for agent communication. If we remove some separators from a CG, the agent communication is effectively restricted to a proper subset of potential paths. The resultant graph is a cluster graph as defined below.

*Definition 7:* Let $(V, \Omega, E)$ be a junction graph and $E' \subseteq E$. Then $(V, \Omega, E')$ is a `cluster` graph over $V$.

Note that a junction graph is also a cluster graph, but a cluster graph may not be a junction graph. Cluster graphs of a CG represents alternative organizations for agent communication, which will be studied in the next section.

## IV. On hypertree organization

### A. Classification of loops

The difficulty of coherent inference in multiply connected graphical models (those with loops) of probabilistic knowledge is well known and many inference algorithms have been proposed. Those based on message passing, e.g., [20], [14], [10], [23], [4], all convert a multiply connected network into a tree. However, *no* formal arguments can be found, e.g., in [20], [9], [18], [3], which demonstrate convincingly that message passing *cannot* be made coherent in multiply connected networks.[3] This leaves the question whether it is *impossible* to construct such a method or the method remains to be discovered, under the constraints that each node in the network is associated with only a local distribution and it is never passed to a central location for manipulation.

The answer to this question ties closely to the necessity of the hypertree organization of agents as specified in Definition 3 and restated as the choice 3 in Section II. This tie can be seen by noting that the hypertree in Definition 3 is isomorphic to a subgraph of the communication graph $H$ of the same multiagent system: A one-to-one mapping exists between hypernodes in Definition 3 and nodes in $H$. Each hyperlink in Definition 3 is a link in $H$ but the converse is not true. Compare Figures 5 and 7 as an example. In what follows, we show that in general, coherent message passing is impossible in general, multiply connected CGs. The result formally establishes the necessity of hypertree structure for uncertain domain state estimation by multiagent message passing. We first classify loops on a cluster graph as follows:

*Definition 8:* Let $H$ be a cluster graph over $V$ and $\rho$ be a loop in $H$. If there exists a separator $S$ on $\rho$ that is contained in every other separator on $\rho$, then $\rho$ is a `degenerate` loop. Otherwise, $\rho$ is a `nondegenerate` loop.

A degenerate loop $\rho$ is a `strong` degenerate loop if all separators on $\rho$ are identical. Otherwise, $\rho$ is a `weak` degenerate loop.

An nondegenerate loop $\rho$ is a `strong` nondegenerate loop if $\cap_i S_i = \emptyset$, where $i$ is over every separator $S_i$ on $\rho$. Otherwise, $\rho$ is a `weak` nondegenerate loop.

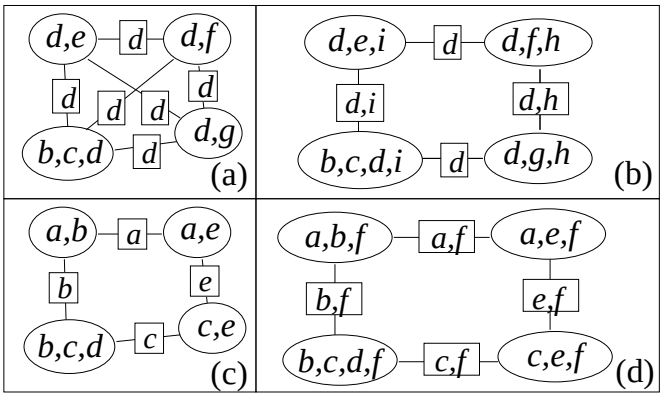



Fig. 8. Cluster graphs where each cluster is shown as an oval and each separator is shown in a box.

In Figure 8, all loops in (a) and (b) are degenerate. The loops in (a) are strong degenerate loops, and the loop in (b) is a weak degenerate loop. The loops in (c) and (d) are nondegenerate. The loop in (c) is a strong nondegenerate loop, and that in (d) is a weak nondegenerate loop. In general, a cluster graph can contain both types of loops and can contain strong and weak loops for each type.

[3] In fact, this issue has never been raised openly to the authors' knowledge. Pearl [20] explained that his $\lambda - \pi$ algorithm for message passing in tree-structured BNs would not work correctly in multiply connected BNs because the assumptions that lead to the algorithm would not hold. He did not, however, treat the issue in general. In fact, an empirical study [17] has been performed recently to apply $\lambda - \pi$ to multiply connected BNs for approximate inference.

### B. Nondegenerate loops

We show that when nondegenerate loops exist, messages are *uninformative.* No matter how messages are manipulated or routed, they cannot become *informative* and it becomes impossible to make message passing coherent.

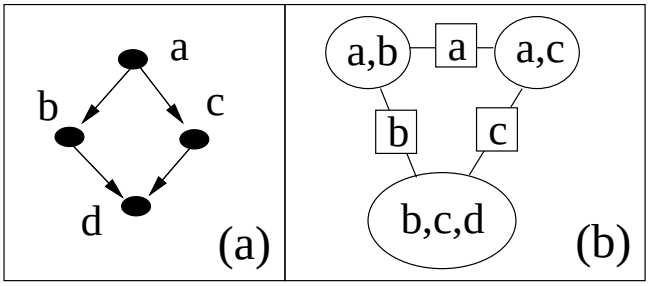



Fig. 9. (a) A dependence structure over four variables. (b) The junction graph for the variables in (a).

Consider a domain with the dependence structure in Figure 9 (a) where $a, b, c, d$ are binary (i.e., $a \in \{a_0, a_1\}$, $b \in \{b_0, b_1\}$, and so on). It is populated by three agents $A_i$ ($i = 0, 1, 2$) with $V_0 = \{a, b\}$, $V_1 = \{a, c\}$ and $V_2 = \{b, c, d\}$. Although the system appears trivial, it will be expanded to arbitrary complexity below. Figure 9 (b) is the communication graph. The local knowledge of agents are $P_0(a, b)$, $P_1(a, c)$ and $P_2(b, c, d)$, respectively. We assume that their beliefs are initially consistent, namely, the marginal distributions satisfy $P_0(a) = P_1(a)$, $P_0(b) = P_2(b)$, and $P_1(c) = P_2(c)$. Due to BC 2, message passing cannot change any agent's belief. We refer to the system as *Mas3* (meaning a multiagent system of 3 agents). Any given $P_0(a, b)$, $P_1(a, c)$ and $P_2(b, c, d)$ subject to the above consistency is called an *initial (belief) state* of *Mas3*.

Suppose that $A_2$ observes $d = d_0$. If the agents can update their beliefs coherently, their new beliefs should be $P_0(a, b|d = d_0)$, $P_1(a, c|d = d_0)$ and $P_2(b, c, d|d = d_0)$. For $A_2$, $P_2(b, c, d|d = d_0)$ can be obtained locally. However, for $A_0$ and $A_1$ to update their beliefs, they must rely on the message $P_2(b|d = d_0)$ sent by $A_2$ to $A_0$ and the message $P_2(c|d = d_0)$ sent by $A_2$ to $A_1$. In the following, we show that $A_0$ and $A_1$ cannot update their beliefs coherently based on these messages. Before the general result, we illustrate with a particular initial state. From Figure 9, we can independently specify $P(a)$, $P(b|a)$, $P(c|a)$, and $P(d|b, c)$ as follows:

$$\begin{array}{lll} P(a_0) = .26 & P(b_0|a_0) = .98 & P(b_0|a_1) = .33 \\ P(c_0|a_0) = .02 & P(c_0|a_1) = .67 & \\ P(d_0|b_0, c_0) = .03 & P(d_0|b_0, c_1) = .66 & \\ P(d_0|b_1, c_0) = .7 & P(d_0|b_1, c_1) = .25 & \end{array}$$

From these, we define an initial state $s$ which is consistent:

$$P_0(a, b) = P(a)P(b|a), \quad P_1(a, c) = P(a)P(c|a),$$

$$P_2(b, c, d) = P(b, c)P(d|b, c),$$

where $P(b, c) = \sum_a P(a)P(b|a)P(c|a)$. After $d = d_0$ is observed by $A_2$, its messages are $P_2(b|d_0) = (0.448, 0.552)$ and $P_2(c|d_0) = (0.477, 0.532)$.

Consider next a different initial state $s'$ that differs from $s$ by replacing $P(d|b, c)$ with $P'(d|b, c)$ as follows:

$$\begin{array}{ll} P'_2(d_0|b_0, c_0) = 0.5336 & P'_2(d_0|b_0, c_1) = 0.1154 \\ P'_2(d_0|b_1, c_0) = 0.14 & P'_2(d_0|b_1, c_1) = 0.66 \end{array}$$

Note that $P'_2(b, c, d) \neq P_2(b, c, d)$, but $P'_0(a, b) = P_0(a, b)$ and $P'_1(a, c) = P_1(a, c)$. After $d = d_0$ is observed, if the messages $P'_2(b|d_0)$ and $P'_2(c|d_0)$ are computed, they are found to be identical to those obtained from the state $s$. That is, the messages are insensitive to the difference between the two initial states. As the consequence, the new beliefs in $A_0$ and $A_1$ will be identical in both cases. Should the new beliefs in both cases be different? Using the coherent probabilistic inference, the new belief $P(a_1|d_0) = 0.666$ is obtained from $s$, and $P'(a_1|d_0) = 0.878$ is obtained from $s'$. The difference is significant.

We now show that the above phenomenon is not accidental. Without losing generality, we assume that all distributions are strictly positive. Lemma 9 says that for infinitely many different initial states of agent $A_2$, its messages to $A_0$ and $A_1$, however, are identical.

*Lemma 9:* Let $s$ be a strictly positive initial state of Mas3. There exists an infinite set $S$. Each element $s' \in S$ is an initial state of Mas3 identical to $s$ in $P(a)$, $P(b|a)$, and $P(c|a)$ but distinct in $P(d|b, c)$ such that the message $P_2(b|d = d_0)$ produced from $s'$ is identical to that produced from $s$, and so is the message $P_2(c|d = d_0)$.

Proof: We denote the message component $P_2(b = b_0|d = d_0)$ from state $s$ by $P_2(b_0|d_0)$. We denote the message component from $s'$ by $P'_2(b_0|d_0)$. $P_2(b_0|d_0)$ can be expanded as

$$P_2(b_0|d_0) = P_2(b_0, d_0)/(P_2(b_0, d_0) + P_2(b_1, d_0))$$

$$= [1 + \tfrac{P_2(b_1,d_0)}{P_2(b_0,d_0)}]^{-1} = [1 + \tfrac{P_2(b_1,c_0,d_0)+P_2(b_1,c_1,d_0)}{P_2(b_0,c_0,d_0)+P_2(b_0,c_1,d_0)}]^{-1}$$

$$= [1 + \tfrac{P_2(d_0|b_1,c_0)P_2(b_1,c_0)+P_2(d_0|b_1,c_1)P_2(b_1,c_1)}{P_2(d_0|b_0,c_0)P_2(b_0,c_0)+P_2(d_0|b_0,c_1)P_2(b_0,c_1)}]^{-1}.$$

Similarly, the message component $P_2(c_0|d_0)$ can be expanded as

$$P_2(c_0|d_0) = [1 + \tfrac{P_2(c_1,d_0)}{P_2(c_0,d_0)}]^{-1}$$

$$= [1 + \tfrac{P_2(d_0|b_0,c_1)P_2(b_0,c_1)+P_2(d_0|b_1,c_1)P_2(b_1,c_1)}{P_2(d_0|b_0,c_0)P_2(b_0,c_0)+P_2(d_0|b_1,c_0)P_2(b_1,c_0)}]^{-1}.$$

By assumption, $P_0(a,b) = P'_0(a,b)$, $P_1(a,c) = P'_1(a,c)$ and $P_2(b,c) = P'_2(b,c)$ but $P_2(d|b,c) \neq P'_2(d|b,c)$. If agent $A_2$ at $s'$ generates the identical messages $P'_2(b|d_0) = P_2(b|d_0)$ and $P'_2(c|d_0) = P_2(c|d_0)$ (conclusion of the lemma), then $P'_2(d|b,c)$ must be the solution of the following equations:

$$\frac{P'_2(d_0|b_1,c_0)P_2(b_1,c_0)+P'_2(d_0|b_1,c_1)P_2(b_1,c_1)}{P'_2(d_0|b_0,c_0)P_2(b_0,c_0)+P'_2(d_0|b_0,c_1)P_2(b_0,c_1)} = \frac{P_2(b_1,d_0)}{P_2(b_0,d_0)}$$

$$\frac{P'_2(d_0|b_0,c_1)P_2(b_0,c_1)+P'_2(d_0|b_1,c_1)P_2(b_1,c_1)}{P'_2(d_0|b_0,c_0)P_2(b_0,c_0)+P'_2(d_0|b_1,c_0)P_2(b_1,c_0)} = \frac{P_2(c_1,d_0)}{P_2(c_0,d_0)}$$

Because $P'_2(d|b,c)$ has four independent parameters but is constrained by only two equations, it has *infinitely* many solutions. Each solution defines an initial state $s'$ of *Mas3* that satisfies all conditions in the lemma. □

Lemma 10 says that with the same difference in initial states, a coherent inference will produce distinct results from *Mas3*.

*Lemma 10:* Let $P$ and $P'$ be strictly positive probability distributions over the DAG of Figure 9 such that they are identical in $P(a)$, $P(b|a)$ and $P(c|a)$ but distinct in $P(d|b,c)$. Then $P(a|d = d_0)$ is distinct from $P'(a|d = d_0)$ in general.

Proof: The following can be obtained from $P$ and $P'$:

$$P(a|d_0) = \sum_{b,c} P(a|b,c)P(b,c|d_0) \quad (1)$$

$$P'(a|d_0) = \sum_{b,c} P(a|b,c)P'(b,c|d_0) \quad (2)$$

where $P(a|b,c)$ is used because $P'$ is identical with $P$ in $P(a)$, $P(b|a)$ and $P(c|a)$. If $P(b,c|d_0) \neq P'(b,c|d_0)$ (which we shall show below), then in general $P(a|d_0) \neq P'(a|d_0)$. We have

$$P(b,c|d_0) = \frac{P(d_0|b,c)P(b,c)}{P(d_0)} = \frac{P(d_0|b,c)P(b,c)}{\sum_{b,c} P(d_0|b,c)P(b,c)},$$

$$P'(b,c|d_0) = \frac{P'(d_0|b,c)P(b,c)}{P'(d_0)} = \frac{P'(d_0|b,c)P(b,c)}{\sum_{b,c} P'(d_0|b,c)P(b,c)}.$$

Because $P(d|b,c) \neq P'(d|b,c)$, in general, it is the case that $P(b,c|d_0) \neq P'(b,c|d_0)$. □

We conclude with the following theorem:

*Theorem 11:* Message passing in Mas3 cannot be coherent in general, no matter how it is performed.

Proof: By Lemma 9, $P_2(b|d = d_0)$ and $P_2(c|d = d_0)$ are insensitive to the initial states and hence the posteriors (e.g., $P_0(a|d = d_0)$) computed from the messages cannot be sensitive to the initial states either. However, by Lemma 10, the posteriors should be different in general given different initial states. Hence, correct belief updating cannot be achieved in *Mas3*. □

Note that the non-coherence of *Mas3* is due to its non-degenerate loop. From Eqs.(1) and (2), correct inference requires $P(b,c|d_0)$. To pass such a message, a separator must contain $\{b,c\}$, the intersection between $U_2$ and $U_0 \cup U_1$. The nondegenerate loop signifies the splitting of such a separator (into separators $\{b\}$ and $\{c\}$). The result is the passing of marginals of $P(b,c|d_0)$ (the insensitive messages) and ultimately the incorrect inference.

We can generalize this analysis to an arbitrary, strong nondegenerate loop of length 3 (the loop length of *Mas3*), where each of $a,\ b,\ c,\ d$ is a set of variables. The result in Lemmas 9, 10 and Theorem 11 can be similarly derived.

We can further generalize this analysis to an arbitrary, strong nondegenerate loop of length $K > 3$. By clumping $K-2$ adjacent subdomains into one big subdomain $Q$, the loop is reduced to length 3. Any message passing among the $k-2$ subdomains can be considered as occurring in the same way as before the clumping but "inside" $Q$. Now the above analysis for an arbitrary strong nondegenerate loop of length 3 applies.

Furthermore, the result can be generalized to an arbitrary, weak nondegenerate loop $\rho$ of length $K \geq 3$. Let

$$\rho = \langle Q_0, Q_1, ..., Q_{K-1}, Q_0 \rangle,$$

the separator between $Q_i$ and $Q_{i+1}$ be $S_i$ $(0 \leq i < K-1)$, the separator between $Q_{K-1}$ and $Q_0$ be $S_{K-1}$, and $R = \cap_{i=0}^{K-1} S_i$. Let the potential of each cluster be

$$B_{Q_i}(Q_i) = P_{Q_i}(Q_i \setminus R|R)P_{Q_i}(R).$$

$P_{Q_i}(R)$ and $P_{Q_j}(R)$ may be different due to an observation on $r \in R$ available to $Q_i$ but not to $Q_j$. This is possible when the object associated with $r$ is *physically* located with the agent associated with $Q_i$ and $r$ is only *logically* known to the agent associated with $Q_j$. Message passing in $\rho$ can be considered as independently passing of two message streams, one determined by $P_{Q_i}(Q_i \setminus R|R)$ and one determined by $P_{Q_i}(R)$. For example, the message from $Q_0$ to $Q_1$ is

$$B_{Q_0}(S_0) = P_{Q_0}(S_0 \setminus R|R)P_{Q_0}(R),$$

where $P_{Q_0}(S_0 \setminus R|R)$ can be obtained by marginalization of $P_{Q_0}(Q_0 \setminus R|R)$.

The first message stream according to $P_{Q_i}(Q_i \setminus R|R)$ is equivalent to the message passing in a strong nondegenerate loop

$$\rho' = \langle Q'_0, Q'_1, ..., Q'_{K-1}, Q'_0 \rangle,$$

where each cluster $Q'_i = Q_i \setminus R$. According to the above analysis, belief updating cannot be achieved by message passing in $\rho'$. The second message stream relative to

$P_{Q_i}(R)$ is straightforward. It in general has no impact on the first message stream. Therefore, belief updating cannot be achieved by message passing in $\rho$ in general.

To summarize, the difficulty will arise whenever a cluster graph contains nondegenerate loops, whether they are strong or weak. This is stated in the following Corollary.

*Corollary 12:* Message passing in a cluster graph with nondegenerate loops cannot be coherent in general, no matter how it is performed.

### C. Degenerate loops

In a strong degenerate loop, all subdomains share the same separator and it is straightforward to pass the message coherently. Furthermore, the coherent message passing can be performed with any one separator omitted. That is, it can be performed with the loop cut open into a chain.

Next, consider a weak degenerate loop $\rho$ where separators are not all identical, but there exists a separator $S$ that is contained in each other separator. Let the loop be

$$\rho = \langle Q_0, Q_1, ..., Q_{K-1}, Q_0 \rangle,$$

where $K \geq 3$. Let the clusters connected by $S$ be $Q_0$ and $Q_1$. There are two paths between $Q_0$ and $Q_1$,

$$\langle Q_0, Q_1 \rangle \ \ and \ \ \langle Q_0, Q_{K-1}, Q_{K-2}, ..., Q_2, Q_1 \rangle.$$

The message that can be passed from $Q_0$ to $Q_1$ along $S$ is a potential $B_{Q_0}(S) = P_{Q_0}(S)$. Because the message passed along any separator $S' \neq S$ can be expressed as a potential

$$B(S') = P(S' \setminus S|S)P(S),$$

which contains $P(S)$, the path $\langle Q_0, Q_1 \rangle$ is *redundant*: the same information can be propagated through the other path. Therefore, whether or not coherent message passing is achievable in a weak degenerate loop can be determined using the cluster chain obtained by breaking the loop at $S$. For example, whether coherent message passing is achievable in the cluster graph in Figure 10 (a) can be determined by deleting the separator $\{v\}$ to obtain (b). Similarly, whether coherent message passing is achievable in the cluster graph in Figure 11 (a) can be determined by deleting a separator $\{u\}$ to obtain (b).

Coherent message passing is achievable in Figure 10 (b) using any well-known methods [14], [9], [24]. Hence, it is also achievable in (a) because one can always ignore the existence of the separator that is omitted in (b). On the other hand, coherent message passing is *not* achievable in general in Figure 11 (b) as exemplified in Figure 12. The potential of each cluster is shown in terms

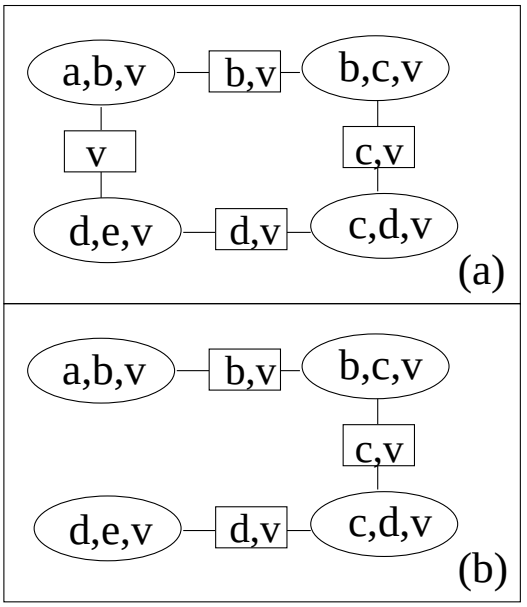



Fig. 10. The weak degenerate loop in (a) is broken into a chain in (b).

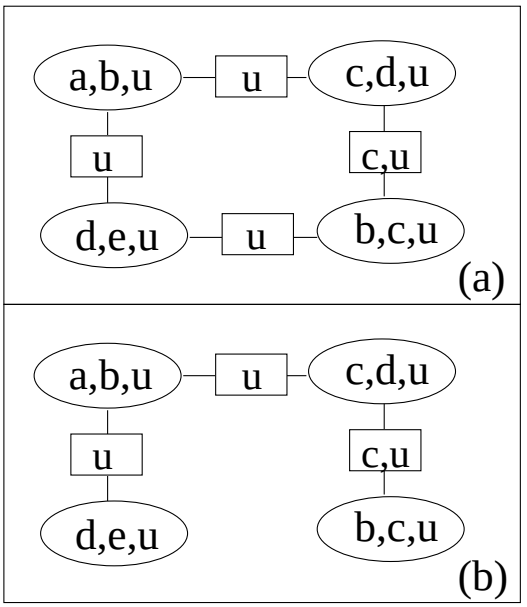



Fig. 11. The weak degenerate loop in (a) is broken into a chain in (b).

of a single non-zero probability value (with zero being the value of other probabilities). Clearly, the message on the separator $\{u\}$ is $P(u = 1) = 1.0$ for each corresponding cluster. The message on the separator $\{c, u\}$ is $P(c = 1, u = 1) = 1.0$. Therefore, no matter how message passing is performed, none of the cluster potentials will change. However, according to the cluster $\{c, d, u\}$, we have $P(d = 1) = 1.0$, but according to the cluster $\{d, e, u\}$, we have $P(d = 1) = 0$. Because the separator between the clusters $\{d, e, u\}$ and $\{b, c, u\}$ in (a) cannot help solve the problem, coherent message passing is also not achievable in Figure 11 (a).

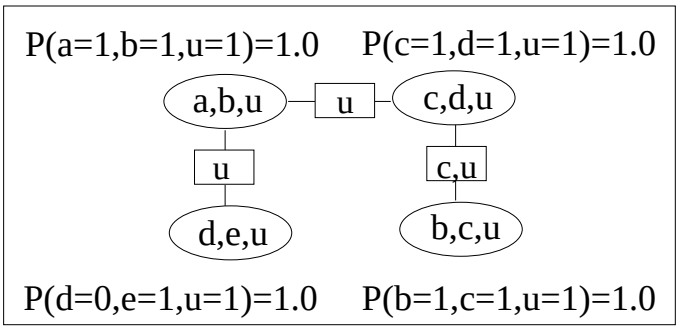



Fig. 12. Each pair of adjacent clusters are consistent.

The key conclusion here is that the loopy structure of a weak degenerate loop is *insignificant* just as that of a strong degenerate loop, in the sense that whether it provides support to coherent message passing can be studied reliably from a derived chain structure. Hence, a cluster graph with only degenerate loops can always be treated by first breaking the loops at appropriate separators. The resultant is a cluster tree.

With the understanding of the properties of different types of loops, we now make a choice on the organizational structure for agent communication. Given any connected graph $G$, its connected spanning subgraphs (containing the same set of nodes as $G$) with the minimum number of links are trees. That is, trees are the simplest (with the minimum number of links) subgraphs that retain connectedness. Simplicity is conductive to efficiency. Consider a weak degenerate loop in a CG, where a separator $S$ is contained in every other separator. If we use the loopy communication organization, there are two information channels between any two clusters in the loop: one through $S$ and one through the other path in the loop. Because each separator in the other path is a superset of $S$ (by definition of weak degenerate loop), from BC 2, the information capacity of the path through $S$ is inferior to the other path. This implies that not all messages can be passed equivalently from both paths. Hence, agents must select the path carefully depending on the content of their messages. Clearly, this requires more sophisticated computation and coordination than what would be required in a tree organization. We therefore prefer a simpler organization of agents when degenerate loops exist in the CG:

*BC 3:* A simpler agent organization (as a subgraph of the communications graph) is preferred.

From BC 3, a tree organization follows. This is summarized in the following proposition, which implies the choice 3 in Section II.

*Proposition 13:* Let a multiagent system be one that observes BC 1 through BC 3. Then a tree organization of agents should be used.

Proposition 13 admits many tree organizations. Jensen [9] showed that coherent message passing may not be achieved with just any tree. In particular, if two subdomains $V_i$ and $V_j$ share a subset $I$ of variables but $I$ is not contained in every subdomain on the path between them in the tree, then coherent message passing is not achievable. In fact, the cluster tree in Figure 11 (b) suffers precisely this problem. To ensure coherent message passing, the tree must be a *junction tree*, where for each pair of $V_i$ and $V_j$, $V_i \cap V_j$ is contained in every subdomain on the path between $V_i$ and $V_j$. Note the similarity between a junction tree and a hypertree in Definition 1. Hence, we have the following proposition:

*Proposition 14:* Let a multiagent system be one that observes BC 1 through BC 3. Then a junction tree organization of agents must be used.

## V. On subdomain separators

Given the commitment to a (hyper) junction tree organization, it follows that each separator must be chosen such that the message over it is *sufficient* to convey all the relevant information from one subtree to the other. Let $Z$ denote the set of variables in the separator, $X$ denote the union of all subdomains of one subtree induced by the separator excluding $Z$, and $Y$ denote the union of all subdomains of the other subtree excluding $Z$. By BC 2, $P(Z)$ is the only information that can be directly communicated between $X$ and $Y$. Note that because we are concerned with $Z$ as the separator between $X$ and $Y$, we can safely ignore the fact that $X$ (or $Y$) is distributed among multiple agents.

We consider the condition under which the messages between $C = X \cup Z$ and $Q = Y \cup Z$ through $Z$ are sufficiently informative to ensure coherent message passing. Suppose $C$ is associated with a potential $P_C(X,Z) = \sum_Y P(X,Y,Z)$ and $Q$ with $P_Q(Y,Z) = \sum_X P(X,Y,Z)$. For the disjoint sets $X$, $Y$, and $Z$ of variables, denote $I(X,Z,Y)$ [20] if $X$ and $Y$ are *conditionally independent* given $Z$. Using the notation, if $I(X,Z,Y)$ holds, then the joint distribution $P(X,Y,Z)$ and local distributions $P_C(X,Z)$, $P_Q(Y,Z)$ satisfy

$$P(X,Y,Z) = P_C(X,Z)P_Q(Y,Z)/P_C(Z),$$

where $P_C(Z) = \sum_X P_C(X,Z) = \sum_Y P_Q(Y,Z) = P_Q(Z)$. Now suppose some variables in $C$ are observed and $P_C(X,Z)$ is updated into $P_C(X,Z|obs)$. To update $P_Q(Y,Z)$ in $Q$, pass the message $P_C(Z|obs) = \sum_X P_C(X,Z|obs)$ from $C$ to $Q$, and replace $P_Q(Y,Z)$ in $Q$ by

$$P_Q(Y,Z|obs) = P_Q(Y|Z) * P_C(Z|obs).$$

The message passing is coherent because

$$P(X,Y,Z|obs) = \frac{P_C(X,Z|obs)P_Q(Y,Z|obs)}{P_C(Z|obs)}.$$

What if $I(X,Z,Y)$ does not hold? Consider the dependence structure in Figure 9 (a). If a Bayesian network is defined with the dependence structure, and $P(a,b,c,d)$ is constructed by the *chain rule* as

$$P(a,b,c,d) = P(a)P(b|a)P(c|a)P(d|b,c),$$

then in general, $I(a,b,\{c,d\})$ does not hold. It has been shown in Lemmas 9, 10 and Theorem 11 that passing a message over $b$ from cluster $\{b,c,d\}$ to $\{a,b\}$ cannot produce correct posterior in general. The following proposition summerizes the above analysis.

*Proposition 15:* Let $X$, $Y$ and $Z$ be disjoint sets of variables with $P(X,Y,Z)$ defined. Let $C = X \cup Z$ be associated with $P_C(X,Z) = \sum_Y P(X,Y,Z)$, and $Q = Y \cup Z$ be associated with $P_Q(Y,Z) = \sum_X P(X,Y,Z)$.

1) If $I(X,Z,Y)$ holds, then message passing can be performed coherently by passing a potential over $Z$ between $C$ and $Q$.
2) If $I(X,Z,Y)$ does not hold, then message passing cannot be performed coherently in general by passing a potential over $Z$ between $C$ and $Q$.

To conclude, when the separator renders the two subtrees conditionally independent, if new observations are obtained in one subtree by the corresponding agents, coherent belief update of agents in the other subtree can be achieved by simply passing the updated distribution on the separator. On the other hand, if the separator does not render the two subtrees conditionally independent, passing only the separator distribution will not be coherent in general. Hence, we have the following proposition:

*Proposition 16:* Let a multiagent system be one that observes BC 1 through BC 3. Then each separator in a tree organization must render subdomains in the two induced subtrees conditionally independent.

This commitment requires the problem domain to be partitioned among agents such that intersections of subdomains form conditional independent separators in a hypertree organization.

## VI. Choice on subdomain representation

Given a subdomain $V_i$, the number of parameters to represent the belief of $A_i$ through a potential over $V_i$ is exponential on the cardinality $|V_i|$. Graphical models allow such belief to be compactly represented. We focus on DAG models as they are the most concise, with the understanding that other models such as decomposable Markov networks [20], [13], [34] or chain graphs [13] may also be used. This corresponds to the choice 4 of Section II.

*BC 4:* A DAG is used to structure each individual agent's knowledge.

A DAG model admits an asymmetric and acyclic interpretation of dependence. Once we adopt it for each agent, we must adopt it for the joint belief of all agents:

*Proposition 17:* Let a multiagent system over $V$ be constructed following BC 1 through BC 4. Then each subdomain $V_i$ is structured as a DAG over $V_i$ and the union of these DAGs is a connected DAG over $V$.

Proof: If the union of subdomain DAGs is not a DAG, then it has a directed loop. This contradicts the acyclic interpretation of dependence in individual DAG models. The connectedness is implied by Proposition 6. □

The choice 5 of Section II now follows.

## VII. On interface between subdomains

We show that the interface between subdomains must be structured as a d-sepset (Definition 2). This is established below through the concept of d-separation [20].

*Theorem 18:* Let $\Psi$ be a hypertree over a directed graph $G = (V,E)$. For each hyperlink $I$ which splits $\Psi$ into two subtrees over $U \subset V$ and $W \subset V$ respectively, $U \setminus I$ and $W \setminus I$ are d-separated by $I$ if and only if each hyperlink in $\Psi$ is a d-sepset.

Before proving the theorem, we explain its rational and importance. Proposition 16 states that each separator in a tree organization must render subdomains in the two induced subtrees conditionally independent. Because d-separation captures all graphically identifiable conditional independencies [20], Theorem 18 implies that d-sepset is the necessary and sufficient *syntactic* condition to ensure conditionally independent separators. We prove Theorem 18 below:

Proof:

[Sufficiency] Assume that each hyperlink is a d-sepset. We show that for any given hyperlink $I$, $U \setminus I$ and $W \setminus I$ are d-separated by $I$.

Let $\rho$ be a path between $U \setminus I$ and $W \setminus I$ such that all nodes in one side of $\rho$ belong to $U \setminus I$, all nodes in the other side belong to $W \setminus I$, and one or more adjacent nodes in $I$ are in between. It suffices to show that every such path is blocked by $I$. Every $\rho$ has at least one d-sepnode. If one d-sepnode on $\rho$ is tail-to-tail or head-to-tail, then $\rho$ is blocked by $I$.

Consider the case where $\rho$ has only one d-sepnode $x$. We show that $x$ cannot be head-to-head on $\rho$. Suppose

that $x$ is head-to-head with parents $y$ and $z$ on $\rho$. Because $x$ is the only d-sepnode on $\rho$, neither $y$ nor $z$ is shared by $U$ and $W$, say, $y \in U$ and $z \in W$. This means $\{y, z\} \not\subset U$ and $\{y, z\} \not\subset W$. Because $x$ is a d-sepnode, there exists a subgraph $G_k$ that contains $\pi(x)$. Because $G_k$ is either located in the subtree over $U$ or the subtree over $W$, either $\pi(x) \subset U$ or $\pi(x) \subset W$ holds. Given $\{y, z\} \subseteq \pi(x)$, it follows that either $\{y, z\} \subset U$ or $\{y, z\} \subset W$ must hold: a contradiction. Hence, $x$ is either tail-to-tail or head-to-tail on $\rho$.

Next, consider the case where $\rho$ contains at least two d-sepnodes. We show that one of them cannot be head-to-head on $\rho$. Pick two d-sepnodes $x$ and $y$ on $\rho$ that are adjacent. Such $x$ and $y$ do exist according to how $\rho$ is defined. The $x$ and $y$ are connected either by $(x, y)$ or by $(y, x)$. In either case, one of them must be a tail node.

[Necessity] Assume that every hyperlink d-separates the two subtrees. We show that each hyperlink is a d-sepset by contradiction.

Suppose that there exists a shared node $x$ such that no subgraph contains $\pi(x)$ (hence not every hyperlink is a d-sepset). Then there exists a hyperlink $I$ on $\Psi$ where $x \in I$, and there exist nonempty subsets

$$\pi_U(x) \subset \pi(x) \ \ and \ \ \pi_W(x) \subset \pi(x)$$

such that $\pi_U(x) \subset U$, $\pi_W(x) \subset W$, $\pi_U(x) \cup \pi_W(x) = \pi(x)$, and $\pi_U(x)$ is incomparable with $\pi_W(x)$. Because $\pi_U(x)$ is incomparable with $\pi_W(x)$, there exist $y \in \pi_U(x)$ but $y \notin \pi_W(x)$, and $z \in \pi_W(x)$ but $z \notin \pi_U(x)$. The path $\rho = \langle y, x, z \rangle$ between $U$ and $W$ is rendered open by $I$ because $x$ is head-to-head on $\rho$. Hence, $U \setminus I$ and $W \setminus I$ are not d-separated by $I$: a contradiction. □

Theorem 18 implies that d-sepset is the necessary and sufficient syntactic condition for conditionally independent separators under *all possible subdomain structures and observation patterns*. We emphasize that d-sepset is necessary for the most general case, because by restricting subdomain structures (e.g., some agent contains only "cause" relative to other agents but no "effect") or observation patterns (e.g., some agent has no local observation and only relies on others' observations), the d-sepset requirement may be relaxed. The choice 6 of Section II now follows. From Propositions 14 and 17, and Theorem 18, the following proposition is implied.

*Proposition 19:* Let a multiagent system be constructed following BC 1 through BC 4. Then it must be structured as a hypertree MSDAG.

Proof:

From BC 1 through BC 4, it follows that each subdomain should be structured as a DAG and the entire domain should be structured as a connected DAG (Proposition 17). The DAGs should be organized into a hypertree (Proposition 14). The interface between adjacent DAGs on the hypertree should be a d-sepset (Theorem 18). Hence, the multiagent system should be structured as a hypertree MSDAG (Definition 3). □

## VIII. On belief assignment

By Propositions 17, the structure of a multiagent system is a connected DAG. Hence, a joint probability distribution (jpd) over the entire domain can be defined by specifying a local distribution for each node and applying the chain rule. In a multiagent system, a node can be internal to an agent or shared by two or more agents. The distribution for an internal node can be specified by the corresponding agent designer. When a node is shared, it may have different parents in different agents (e.g., $z_4$ in Figure 2 and figure3). Because each shared node is a d-sepnode, Definition 2 implies that for each shared variable $x$, there exists a subdomain containing all the parents of $x$ in the entire domain as stated in the following lemma:

*Lemma 20:* Let $x$ be a d-sepnode in a hypertree MSDAG $G = \cup G_i$. Let the parents of $x$ in $G_i$ be $\pi_i(x)$. Then there exists $G_k$ such that $\pi_k(x) = \bigcup_i \pi_i(x)$.

If agents are built by the same designer, then once $P(x|\pi_k(x))$ is specified for $x$, $P(x|\pi_i(x))$ for each $i$ is implied. If agents are built by different designers, then it is possible that distributions for a d-sepnode at different subnets may be incompatible with one another. For instance, in Figures 2 and 3, $A_1$ and $A_2$ may differ on $P(g_7)$. We make the following basic commitment for integrating independently built agents into a multiagent system:

*BC 5:* Within each agent's subdomain, the jpd is consistent with the agent's belief. For shared nodes, the jpd supplements each agent's knowledge with others'.

The key issue is to combine agents' belief on a shared variable to arrive at a common belief. One idea [21] is to interpret the distribution from each agent as obtained from a sample data. The combined $P(x|\pi(x))$ can then be obtained from the combined data sample. In summary, let agents combine their belief for each shared $x$. Then, for each shared $x$, let jpd be consistent with $P(x|\pi_k(x))$, and for each internal $x$, let jpd be consistent with $P(x|\pi(x))$ held by the corresponding agent. It's easy to see that the resultant jpd is precisely the one defined in Definition 4, stated in the following proposition:

*Proposition 21:* Let a multiagent system be constructed following BC 1 through BC 5. Then the jpd over $V$ is identical to that of Definition 4.

The last choice of Section II now follows. Pooling Propositions 19 and 21 together, the MSBN representation is entailed by the BCs:

*Theorem 22:* Let a multiagent system be constructed following BC 1 through BC 5. Then it must be represented as an MSBN or some equivalent.

Before concluding this section, we emphasize that the belief consistency between agents that is required by BC 5 concerns only the *shared* variables and concerns only the *background* or *prior* knowledge of agents about these variables. An agent's belief on *private* variables are not constrained directly by the beliefs of any other agents (although it will be influenced by what other agents have observed). For the shared variables, BC 5 requires only that the agents reach an agreement on the *prior* belief. At run time, due to local observations, agents' beliefs on shared variables can become inconsistent. Bringing their beliefs back to consistency will be achieved by agent communication [29].

## IX. Conclusion

From the following *basic* commitments: [BC 1] exact probabilistic measure of belief, [BC 2] communication by belief over small sets of shared variables, [BC 3] a simpler organization of agents, [BC 4] DAG for domain structuring, [BC 5] joint belief admitting agents' beliefs on internal variables and combining their beliefs on shared variables, we have shown that the resultant representation of a cooperative multiagent system is an MSBN or some equivalent.

This result aids comparison with related frameworks. Multiagent inference frameworks based on default reasoning (e.g., DATMS [16] and DTMS [8]) do not admit BC 1, nor does the *blackboard* [19]. The BDI architecture [22] has been very influential in building multiagent systems. It primarily deals with representation of an agent's mental state for practical reasoning [26] although it has been extended to deal with communications between agents [7]. Several frameworks for decomposition of probabilistic knowledge have been proposed. *Abstract network* [12] replaces fragments of a centralized BN by *abstract arcs* to improve inference efficiency. *Similarity network* and *Bayesian multinet* [6] represent asymmetric independence where each subnet shares almost all variables with each other subnet. A *nested junction trees* [11] can exploit independence induced by incoming messages to a cluster and it shares all its variables with the nesting cluster. They were not intended for multiagent systems and do not admit BC 2. Among these alternative frameworks, MSBNs are unique in satisfying both BC 1 and BC 2 in one framework.

Junction tree based message passing algorithms, e.g., [9], [24], [14], like the above mentioned frameworks for probabilistic reasoning with graphical models, are not intended for multiagent systems. However, one might interpret a cluster in a junction tree as corresponding to an agent and its subdomain. Under such an interpretation, a junction tree representation satisfys BC 1 and BC 2. However, a cluster corresponds to a completely connected set of variables. There is no internal structure and the agent's belief is essentially represented in terms of a joint probability distribution over its subdomain. Clearly, both local inference in an agent and communication among agents will be intractable. Hence a junction tree representation under a multiagent interpretation does not admit BC4. On the other hand, MSBNs allow an agent's internal knowledge to be encoded as a Bayesian subnet. This allows both local inference within an agent as well as communication to be performed efficiently (when the subnets are sparse).

This analysis addresses issues on representational constraints required by MSBNs. In particular, the two key technical constraints, hypertree and d-sepset interface, are the consequence of BC 1 and BC 2. Efficient methods for verifying these constraints in a multiagent system have been developed [32], [30].

One useful consequence of BC 2 and the MSBN framework is that the internal knowledge of each agent is never transmitted and can remain private. This aids construction of multiagent systems by agents from independent designers. Multiagent systems commonly stand in two extremes: self-interested versus cooperative. MSBNs stand in the middle: agents are cooperative and truthful to each other while the internal know-how is protected.

Reasoning and acting in uncertain domains are essential issues for multiagent systems. A recent trend has focused on modeling using Markov decision processes (MDP) [2], [35]. It has been shown [1] that the computation for solving distributed MDPs is intractable. Hence, heuristics and approximation must be applied. On the other hand, probabilistic inference in sparse MSBNs is distributed, exact, and efficient [29]. Therefore, extending MSBNs to probabilistic reasoning and decision making over extended time period (as dynamic Bayesian networks [5] extend BNs) provides an alternative representation to the distributed MDP approach. The distributed MDP approach can be viewed as extending the centralized MDP to multiagent systems. The alternative approach can be viewed as extending multiagent uncertain reasoning from static domains to dynamic domains. The result presented in this paper highlights the role of MSBNs in exploring the alternative approach.

Furthermore, our analysis provides guidance to extensions and relaxations of the MSBN framework. Less fundamental constraints can be relaxed, e.g., BC 4 so that

other graph models can be used. BC 3 requires that degenerate loops be broken open in the agent organization. If flexibility in agent communication paths are highly desired, the analysis shows that loopy organizations can be used (with a cost in efficiency) when the loops are certain types of degenerate loops. If subdomain structures and observation patterns are less than general, the d-sepset restriction can be relaxed.

## ACKNOWLEDGEMENTS

This work is supported by Research Grant OGP0155425 from NSERC of Canada, by NSF under Grant No. IIS-9812755, and by DARPA and Air Force Research Lab under F30602-99-2-0525.

## REFERENCES

[1] D.S. Bernstein, S. Zilberstein, and N. Immerman. The complexity of decentralized control of Markov decision processes. In *Proc. 16th Conf. on Uncertainty in Artificial Intelligence*, Stanford, 2000.

[2] C. Boutilier. Multiagent systems: Challenges and opportunities for decision-theoretic planning. *AI Magazine*, pages 35–43, Winter, 1999.

[3] E. Castillo, J. Gutierrez, and A. Hadi. *Expert Systems and Probabilistic Network Models*. Springer, 1997.

[4] A. Darwiche. Recursive conditioning. *Articial Intelligence*, 126(1-2):5–41, 2000.

[5] T.L. Dean and K. Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, (5):142–150, 1989.

[6] D. Geiger and D. Heckerman. Knowledge representation and inference in similarity networks and Bayesian multinets. *Artificial Intelligence*, 82:45–72, 1996.

[7] A. Haddadi. Communication and cooperation in agent systems. In *LNAI Volume 1056*. Springer, Berlin, 1996.

[8] M.N. Huhns and D.M. Bridgeland. Multiagent truth maintenance. *IEEE Trans. Sys., Man, and Cybernetics*, 21(6):1437–1445, 1991.

[9] F.V. Jensen. *An Introduction To Bayesian Networks*. UCL Press, 1996.

[10] 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.

[11] U. Kjaerulff. Nested junction trees. In *Proc. 13th Conf. on Uncertainty in Artificial Intelligence*, pages 294–301, Providence, Rhode Island, 1997.

[12] W. Lam. Abstraction in Bayesian belief networks and automatic discovery from past inference sessions. In *Proc. of AAAI*, pages 257–262, 1994.

[13] S.L. Lauritzen. *Graphical Models*. Oxford, 1996.

[14] S.L. Lauritzen and D.J. Spiegelhalter. Local computation with probabilities on graphical structures and their application to expert systems. *J. Royal Statistical Society, Series B*, (50):157–244, 1988.

[15] V.R. Lesser and L.D. Erman. Distributed interpretation: a model and experiment. *IEEE Trans. on Computers*, C-29(12):1144–1163, 1980.

[16] C.L. Mason and R.R. Johnson. DATMS: a framework for distributed assumption based reasoning. In L. Gasser and M.N. Huhns, editors, *Distributed Artificial Intelligence II*, pages 293–317. Pitman, 1989.

[17] K.P. Murphy, Y. Weiss, and M.I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In K.B. Laskey and H. Prade, editors, *Proc. 15th Conf. on Uncertainty in Artificial Intelligence*, pages 467–475, Stockholm, 1999.

[18] R.E. Neapolitan. *Probabilistic Reasoning in Expert Systems*. John Wiley and Sons, 1990.

[19] H.P. Nii. Blackboard systems: the blackboard model of problem solving and the evolution of blackboard architectures. *AI Magazine*, 7(2):38–53, 1986.

[20] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[21] D. Poole, A. Mackworth, and R. Goebel. *Computational Intelligence: A Logical Approach*. Oxford University Press, 1998.

[22] A. Rao and M. Georgeff. Deliberation and its role in the formation of intentions. In B. D'Ambrosio, P. Smets, and P.P. Bonissone, editors, *Proc. 7th Conf. on Uncertainty in Artificial Intelligence*, pages 300–307. Morgan Kaufmann, 1991.

[23] R.D. Shachter, B. D'Ambrosio, and B.A. Del Favero. Symbolic probabilistic inference in belief networks. In *Proc. 8th Natl. Conf. on Artificial Intelligence*, pages 126–131, 1990.

[24] G. Shafer. *Probabilistic Expert Systems*. Society for Industrial and Applied Mathematics, Philadelphia, 1996.

[25] K.P. Sycara. Multiagent systems. *AI Magazine*, 19(2):79–92, 1998.

[26] M. Wooldridge. *An Introduction to Multiagent Systems*. John Wiley & Sons, 2002.

[27] M. Wooldridge and N.R. Jennings. Intelligent agents: theory and practice. *Knowledge Engineering Review*, 10(2):115–152, 1995.

[28] Y. Xiang. A probabilistic framework for cooperative multi-agent distributed interpretation and optimization of communication. *Artificial Intelligence*, 87(1-2):295–342, 1996.

[29] Y. Xiang. Belief updating in multiply sectioned Bayesian networks without repeated local propagations. *Inter. J. Approximate Reasoning*, 23:1–21, 2000.

[30] Y. Xiang and X. Chen. Cooperative verification of agent interface. In *Proc. 1st Europen Workshop on Probabilistic Graphical Models*. 2002.

[31] Y. Xiang and H. Geng. Distributed monitoring and diagnosis with multiply sectioned Bayesian networks. In *Proc. AAAI Spring symposium on AI in Equipment Service, Maintenance and Support*, pages 18–25, Stanford, 1999.

[32] Y. Xiang, K.G. Olesen, and F.V. Jensen. Practical issues in modeling large diagnostic systems with multiply sectioned Bayesian networks. *Inter. J. Pattern Recognition and Artificial Intelligence*, 14(1):59–71, 2000.

[33] 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.

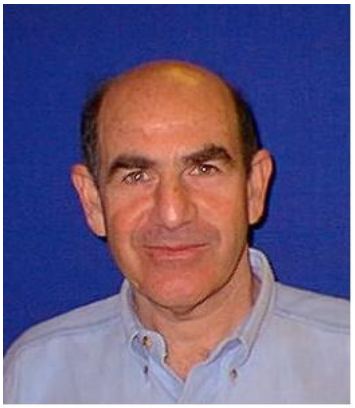
[34] Y. Xiang, S.K.M. Wong, and N. Cercone. A 'microscopic' study of minimum entropy search in learning decomposable Markov networks. *Machine Learning*, 26(1):65–92, 1997.

[35] P. Xuan, V. Lesser, and S. Zilberstein. Communication in multiagent Markov decision processes. In *Proc. ICMAS Workshop on Game Theoretic and Decision Theoretic Agents*, 2000.

## AUTHORS' PHOTO AND BIO

Yang Xiang received the Ph.D. degree from University of British Columbia, Canada in 1992.

He was a Postdoctoral Fellow at Simon Fraser University, Canada, Assistant Professor and then Associate Professor in the Department of Computer Science, University of Regina, Canada, Visiting Professor at Aalborg University, Denmark, and Visiting Professor at University of Massachusetts, Amherst, USA. He is currently an Associate Professor in the Department of Computing and Information Science and Director of Intelligent Decision Support Systems Laboratory, University of Guelph, Canada. His research interests include uncertain reasoning, probabilistic graphical models, multiagent systems, knowledge discovery from data, and business, medicine and industry applications of intelligent systems.



Victor Lesser received his Ph.D. from Stanford University in 1972.

He has been a Professor of Computer Science at the University of Massachusetts at Amherst since 1977. He is a founding fellow of AAAI, and the founding president of the International Foundation for Multi-Agent Systems. His major research focus is on the control and organization of complex AI systems. He has been working in the field of Multi-Agent Systems for over 25 years. Prior to coming to the University of Massachusetts, he was a research scientist at Carnegie-Mellon University where he was the systems architect for the Hearsay-II speech understanding system, which was the first blackboard system developed.