Computational Intelligence, Volume 59, Number 000, 2008

Probabilistic Multiagent Systems and the Rumor Problem—DRAFT

Scott Langevin
Oculus Info Inc., Toronto, Ontario

Marco Valtorta
Department of Computer Science and Engineering, University of South Carolina

We define a cooperative multiagent system where the agents use locally designed Bayesian networks to represent their knowledge. Agents in the newly defined AEBN model communicate via message passing where the messages are beliefs in shared variables that are represented as probability distributions. Messages are treated as soft evidence in the receiver agents, where the belief in the receiving agent is replaced by the publishing agent's belief. We call this the oracular assumption, where one agent is an expert or more knowledgeable of particular variables. As a result, the agents are organized in a publisher-subscriber hierarchy. A central problem of message passing in probabilistic systems is the so called rumor problem, where cycles in message passing cause redundant influence of beliefs. In this paper, algorithms to identify and solve the rumor problem in the context of our multiagent system are presented. The AEBN model is compared and contrasted with the MSBN multiagent model.

We implement several multiagent systems for experimentation using our AEBN multiagent system and the MSBN system and we compare them using several performance measures. From this comparison, we provide guidance for the design of probabilistic multiagent systems.

Key words: Reasoning; Uncertainty; Probabilistic Reasoning; Bayesian Networks; Multiagent Systems.

1. INTRODUCTION

Large real world intelligent systems are often too complex or expensive to build as centralized systems. The computational cost of the large scale reasoning required can be too prohibitive, the scale and scope of the system too complex for a monolithic system, as well aspects of the system are often distributed physically further complicating the construction of a single agent system. To overcome these challenges, the inference and decision making tasks can be decomposed into sub-problems that are reasoned about locally. Multiagent systems can be used to achieve this modular system design, where each agent is responsible for one or more sub-problems. Through agent communication information is exchanged between agents in order to achieve distributed inferencing and decision making.

The reasoning and decision making task often must cope with uncertainty in the problem domain. The uncertainty may come from unobservable aspects of the domain that must be estimated from aspects that are observable, incomplete understanding of the domain, observations that are imprecise, ambiguous, noisy, or unreliable, and lack of resources necessary to observe all relevant events.

Bayesian networks are a probabilistic framework for reasoning with uncertainty. Although Bayesian networks have greatly reduced the time, space and design complexity involved with reasoning using a probability distribution, large complex single networks are challenging to design. Often the computational cost of exact inference is not possible and approximate methods must be employed. To overcome these limitations, what is needed is to divide the network into smaller, manageable

Draft for use by participants in the special session "Recent Developmetn in Graphical Models" at ASMDA-11.

Address correspondence to Marco Valtorta, mgv@cse.sc.edu.

units that are locally reasoned upon and aggregated to solve the global problem. Often this task is described as distributed Bayesian networks.

We provide clear assumptions about agents that use probabilistic representations of knowledge, guidelines for their design, and efficient algorithms for communicating (or sharing) probabilities. Our goal is the design of a cooperative multiagent system where the agents use locally designed Bayesian networks to represent their knowledge. Previous approaches to this problem have imposed strong restrictions on the topology of agent communication, tightly coupled the agents, and have not emphasized the autonomy of each agent. Our agent model attempts to address these deficiencies by loosely coupling the agents and allow for more flexible agent topologies.In this system, agents communicate via message passing where the messages are beliefs in shared variables that are represented as probability distributions. Messages are treated as soft evidence in the receiver agents, where the belief in the receiving agent is replaced by the publishing agent's belief. We call this the oracular assumption, where one agent is an expert or more knowledgeable of particular variables. As a result, the agents are organized in a publisher-subscriber hierarchy. Our model extends the Agent Encapsulated Bayesian Network (AEBN) system proposed by Bloemeke (Bloemeke, 1998). We expand and correct technical details, as well as provide a theoretical basis for AEBNs.

A central problem of message passing in probabilistic systems is the so called rumor problem, where cycles in message passing cause redundant influence of beliefs. We show the solution proposed by Bloemeke is deficient, and develop new algorithms to identify and solve the rumor problem in the context of our multiagent system.

A comparison with other frameworks is presented to highlight the advantages and disadvantages of each, as well as argue for the role of AEBNs in designing probabilistic multiagent systems.

We implemented a multiagent system for experimentation using our multiagent system formalism and compare implementations of the same examples using Xiang's MSBN formalism. The evaluation required the formulation of meaningful performance measures.

The remainder of this paper is organized as follows: Section 2 presents closely related research and introduces our agent model. Section 3 presents the rumor problem in probabilistic systems. Section 4 presents a simulation devised to evaluate our proposed system and scalability analysis. Finally, Section 5 summarizes the contributions and future work of this research.

2. PROBABILISTIC MULTIAGENT SYSTEMS

Pan identifies three main issues of probabilistic multiagent systems (Pan, 2006):

(1) How is a joint probability distribution decomposed amongst the agents?
(2) How are beliefs or local observations exchanged amongst agents?
(3) How is global consistency maintained in the system?

The assumptions and constraints defined for a probabilistic multiagent model to address these three main issues lead to different formalisms with various advantages and disadvantages.

2.1. Multiply Sectioned Bayesian Networks

Multiply Sectioned Bayesian Networks (MSBNs) (Xiang, 2002; Xiang and Lesser, 2000) is a knowledge representation formalism for multiagent uncertain reasoning that effectively sections a large Bayesian network into subnetworks that are each assigned to an agent in the multiagent system.

The MSBN knowledge representation formalism is built up from five guiding basic assumptions of an "ideal" probabilistic multiagent system. These assumptions are used to derive the requirements and constraints necessary that give rise to MSBNs. For discussion on how the assumptions lead to the MSBN formalism see (Xiang, 2002, Chapter 6). The five basic assumptions of MSBNs are:

(1) Each agent's belief is represented by probability.
(2) Agent's communicate with concise messages that are joint probability distributions over the variables they share.
(3) A simpler agent organization is preferred in which agent communication by concise message passing is achievable.

(4) Each agent represents its knowledge dependence structure as a DAG.
(5) Within each agent's subdomain, a JPD is consistent with agent's belief. For shared variables, a JPD supplements an agent's knowledge with others'.

Xiang argues the logical consequence of the basic assumptions is a hypertree structure that is built from a multiply sectioned directed acyclic graph (hypertree MSDAG). Each node in the hypertree represents an agent and the DAG represents the Bayesian network that models the agent's subdomain of knowledge. Thus as opposed to having one distinct, locally designed, Bayesian network that is encapsulated within each agent, each agent effectively contains a piece of a larger, globally designed, Bayesian network. Agents communicate only with neighbors in the hypertree by passing messages that are made up of the variables shared among the agents. The agent interfaces d-separate the subdomains.

A formal set of definitions for MSBNs are presented below using the definitions from Xiang (Xiang, 2002).

Definition 1 (Hypertree MSDAG): Let $G = (V, E)$ be a connected DAG sectioned into subgraphs $\{G_i = (V_i, E_i)\}$. Let the G_i's be organized as a connected tree Ψ, where each node is labelled as G_i and each link between G_k and G_m is labeled by the 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 Ψ (the running intersection property). Then Ψ is a *hypertree* over G. Each link between the subgraphs of Ψ is a *hyperlink*. A *hypertree MSDAG* is a hypertree if each node x contained in more than one subgraph, there exists a subgraph G_i that contains its parents, $\pi(x)$.

Definition 2 (MSBN): An MSBN M is a triplet $(V, G, P) : V = \bigcup_i V_i$ is the total universe where each V_i is a set of variables called a subdomain. $G = \bigcup_i G_i$ (a hypertree MSDAG) is the structure where nodes of each subgraph G_i are labeled by elements of V_i. Let x be a variable and $\pi(x)$ be all 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 subgraphs is assigned a uniform potential. $P = \prod_i P_i$ is the JPD, where each P_i is the product of the potentials associated with nodes in G_i. Each 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 in the hypertree.

Distributed inferencing in a MSBN multiagent system is performed by compiling the hypertree MSDAG into a linked junction forest, which is a tree of junction trees. An inferencing scheme analogous to message passing in junction trees is used to collect and distribute evidence amongst agents, ensuring global consistency in the system.

MSBN was originally proposed for distributed inferencing over a global Bayesian network by assigning subnetworks to individual processors for efficient inferencing on computationally constrained equipment (Xiang, 1992). Xiang argues MSBNs are also appropriate for multiagent systems where the subnetworks can be individually designed as long as *soundness of sectioning* is achievable.

The MSBN framework is appropriate if the multiagent system can be compiled into a linked junction forest. This is a highly restrictive constraint on the multiagent organization and the internal knowledge model of each agent. To compile the agent system into a linked junction forest, the hypertree MSDAG first must perform a distributed moralization and triangulation procedure over all the agents to ensure consistency of the agent's graphical models. The triangulation has a partial order constraint in order to ensure a linkage tree can be constructed, which is a data structure used for efficiency of communication. Once this step is complete, each agent constructs a local junction tree for efficient inferencing over its subdomain. From the local junction trees and the linkage trees, the original hypertree MSDAG is converted into a linked junction forest over which system level inferencing is performed.

MSBNs have the requirement that the union of the local DAGs of agents must also be a DAG. In order to ensure this, a distributed verification process must be performed to ensure the acyclicity of the union of each agent's DAG. If directed cycles exist, the composition of the agents is invalid.

When variables exist in more than one subnet, only one subnet that contains the complete family specifies the conditional distribution of the variable. This is necessary to ensure local and global consistency in the agent system. The method proposed is to assign the "correct" conditional probability table to one agent, and a uniform distribution to all others. Determining the CPT to respect requires intervention from a system designer, or a complex negotiation scheme. However, in individually designed agents, there may not exist an agent that contains the complete family of an interface variable. Xiang devised a distributed algorithm to determine if each interface variable meets this requirement. If this is not the case, the agent decomposition is not valid and agents may need to be merged, or their subdomains modified to satisfy this requirement.

The verification and compilation algorithms demonstrate the complexity involved with probabilistic reasoning in multiagent systems. The complexity is required if one commits to the five basic assumptions as Xiang shows admirably. However, the restrictions imposed by MSBN semantics restrict the autonomy of the agents, and imposes a tight coupling of the agents that limits their applicability in distributed reasoning. Xiang has made several arguments to support the role of MSBNs for multiagent systems (Xiang and Hanshar, 2010; Xiang and Lesser, 2003, 2000; Xiang, 2002, 1996), however, individually designing the agents can be a challenging task (and may not be possible) in order to satisfy the requirements of the MSBN model. Xiang provides little guidance on how to design agents to satisfy the restrictions, rather only methods for checking whether the design is sound.

MSBN is focussed on the distributed computation of a global Bayesian network, where each agent is responsible for a sub-network. In our proposed system we designate agents with expert knowledge over particular variables and the sharing of this knowledge among interested agents. The internal models of each agent are the concern of each agent's designer. This stresses the autonomy, rather than the distributed computational aspects of the multiagent system. We only require consistency over the shared variables to support normative system behavior. Their hidden (non-shared) variables represent an agent's internal beliefs about the world and therefore are not relevant outside the agent.

Our approach is distinct from Xiang's in that we commit to a different set of basic assumptions, resulting in a different formalism. Our model is less complex as it does not require the agents to be organized into a hypertree MSDAG, but instead introduces a strong independence relation called the oracular assumption. Xiang describes MSBNs as a tightly coupled framework (Xiang and Hanshar, 2010); it is our goal to set out to construct a loosely coupled framework that stresses the autonomy of the agents. Additionally, Xiang shows that concise message passing is only achievable in tree topologies. We adopt Xiang's basic assumptions, but we relax basic assumptions #2 and #3 in order to allow agents to be organized in more complex topologies than trees. Further, we introduce a new basic assumption, the oracular assumption, that ensures global consistency is achieved via message passing in multiply connected graphs.

MSBNs are more expressive than our agent model and are more appropriate for agent systems that must coordinate. They also are based on conditional independence relations in the system and do not require additional independence assumptions not present in Bayesian networks, whereas in our model we introduce an oracular assumption. Additionally, since the agents conform to a tree topology, MSBN does not have a rumor problem. Our agent model allows for more flexible topologies than trees, hence we must detect and compensate for rumors to avoid bias. In our framework, agents can be organized as multiply connected graphs, with some restrictions on communication as will be discussed in the next section.

2.2. Agent Encapsulated Bayesian Networks

We now present our agent model, which extends Bloemeke's Agent Encapsulated Bayesian Network model. First, we briefly provide an overview of Bloemeke's work and restrictions we aim to address. Finally, we present a formal description of our extended model.

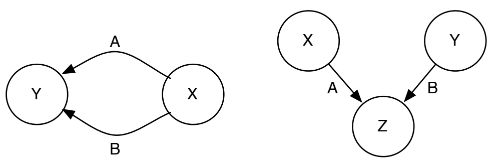
2.2.1. *Original Formalism.* This research extends the Agent Encapsulated Bayesian Network (AEBN) multiagent model originally proposed by Bloemeke (Bloemeke, 1998). This model describes a method of linking individually designed Bayesian networks using a multiagent framework, where each agent utilizes a Bayesian network to represent its internal knowledge representation of the world. The agents communicate by passing messages that are represented as probability distributions over shared variables.

The agents in this model are organized in a publisher-subscriber hierarchy, where the topology of agent communication must conform to a DAG structure. Agents pass messages from publisher to subscriber that are single marginal probability distributions over variables they share. Producer agents are assumed to be experts or more knowledgeable about the variables they produce and share this information with subscribing agents via message passing. Subscribing agents integrate the beliefs of the publishing agents by revising their internal model so it is consistent with the publishing agent. This is done simply by replacing the agent's current view of the shared variable with the publishers. In this way, the publishing agent is said to have oracular knowledge over the variables they produce.

The originally proposed method of revising an agent's Bayesian network relied on the assumption of independence of all received evidence, E. The belief revision was performed using Jeffrey's rule to update $P(V)$ with $Q(E)$:

$$P(V) = \frac{P(V)\prod_{E_i \in E} Q(E_i)}{P(E)}$$

This restriction as well as the limitation of only passing single marginals can result in a loss of dependence among shared variables in two possible ways. We illustrate these two cases with the following two examples. In the first example, consider an agent X sends its beliefs of variables A and B to an agent Y, shown in Figure 1(a). Since only single marginals are passed, agent Y will receive $P(A)$ and $P(B)$ losing any dependence between them.



(a) (b)

Figure 1. Loss of dependence between variables A and B.

In the second example, agent X and Y send their beliefs in variables A and B respectively to agent Z, shown in Figure 1(b). The revision procedure will result in forcing A and B to be independent in agents Z's internal model after absorbing the passed messages. Should we not commit to a revision procedure that minimizes the change in agent Z's model but respects the beliefs of agent X and Y?

Valtorta et al. (Valtorta et al., 2002) argue that agents should not force independence of the received evidence, and illustrate their argument with the following example[1]:

Example 1: Consider an agent that models the age group (A) and education level (E) of US citizens. The agent conducts a small survey on a sample of the population and calculates a joint probability distribution, $P(A, E)$. Later, the agent communicates with two other agents that provide it with accurate US census data for age groups, $Q(A)$, and education levels, $Q(E)$. If we treat the evidence as independent in the receiving agent, then the agent revises its model $P(A, E)$ as $Q(A, E) = Q(A) \cdot Q(E)$. In doing so, it loses all information from its survey!

Instead, Valtorta et al. argue the agent should instead revise its model to a joint probability distribution, $Q^*(A, E)$ such that:

- The marginals for the US census agents are respected.
- The distribution $Q^*(A, E)$ is the distribution that is closest to the original distribution $P(A, E)$ (measured as I-divergence).

Soft evidential update can be used to satisfy these requirements by revising an agent's joint probability distribution to respect the beliefs received from publishing agents.

The independence restrictions limit the expressiveness of AEBNs. We propose an extension of AEBNs to address these problems in the next section.

[1]The example was inspired by Demming and Stephans 1940 paper on IPFP (Deming and Stephan, 1940).

Message passing in multiply connected agent graphs results in the well known rumor problem. Bloemeke proposed a method of identifying redundant influences in an AEBN network, and two methods of compensating for the rumors. The first is a communication solution that extends message passing by passing joint probability distributions. The second is a model-based solution. No formal proofs of correctness of the solutions were provided. Further, the solutions were devised with the assumption of independence of received evidence, and are not appropriate for our agent model. We propose a communication solution that is effective in the context of our agent system and prove its correctness under an assumption of coherence[2].

2.2.2. *Proposed Framework.* In our proposed agent model, each agent represents its internal knowledge base as a Bayesian network. Each agent's probabilistic model is partitioned into three sets of variables:

- Input variables (I): variables which other agents have better knowledge
- Output variables (O): variables which this agent has the best knowledge and that are shared with other agents
- Local or hidden variables (L): variables which are private to this agent

Definition 3 (Agent Encapsulated Bayesian Network): An AEBN is defined as a tuple: $A = (I, L, O, E, P)$, where I is a set of input variables that other agents have better knowledge of, L is a set of local variables and O is a set of output variables that the agent has oracular knowledge of. The union $V = I \cup L \cup O$ define the variables of the AEBNs local Bayesian network, where E is the edges in the model that define the causal relationships amongst the variables V and P is the unique joint probability distribution defined over V. The union $S = I \cup O$ are the AEBNs shared variables, and L are its private (non-shared) variables.

Our desire is to ensure global consistency of shared variables, while minimizing the changes to each agent's local model. This is achieved by treating messages as soft evidence and utilizing soft evidential update.

Each agent provides its best guess as to the correct distribution of its input variables, and relies on other more knowledgable agents providing it with a more accurate view. In the event, no agent can be found, or communication is severed, the overall system will gracefully degrade due to the agents using their estimated guesses or last received belief from the knowledgeable agent.

Agents communicate via passing of messages that are joint probability distributions over their shared output variables, O. The topology of the communication in the multiagent system must conform to a DAG structure to ensure equilibrium can be reached. The agents are organized into a publisher/subscriber hierarchy, where agents are publishers of their output variables and subscribers to their input variables. The underlying assumption is known as the oracular assumption, where one agent is more knowledgable about certain variables and shares its knowledge with interested agents. It is permissible for multiple agents to share knowledge over the same quantities, however, each quantity must have its own unique label.

The method of updating an agent's probability distribution upon the receipt of messages from other agents is described in a related paper (Valtorta et al., 2002), where the messages are called *soft evidence*. In particular, we adopt the modeling approach of introducing *observation variables* into an agents Bayesian network, and updating the agent's probability distribution using the approach of soft evidential update (Valtorta et al., 2002; Chan and Darwiche, 2003). Therefore, each agent that receives messages from other agents obtains soft evidence for one or more observation variables[3] (see Figure 2) that are created by the following procedure:

(1) Create an observation variable, Obs_i, for each soft evidence received, where the states of the observation variable correspond to the possible outcomes of the soft evidence.

[2]An outline of the improved and corrected communication solution has been presented in (Langevin et al., 2010).

[3]The introduction of observation variables is a modeling technique that enables update on a single observation node, rather than a set of nodes.

(2) Add directed edges to Obs_i from all variables in the Bayesian network that have a direct influence on the observation. The set of parents *d-separates* the observation variable from the rest of the network.
(3) Model the logical dependence of the parents of Obs_i, $\pi(Obs_i)$, on Obs_i by specifying the conditional probability table $P(Obs_i|\pi(Obs_i))$ where
$P(Obs_i = o|\pi(Obs_i) = \vec{x}) = 1 \iff \vec{x}$ corresponds to o.

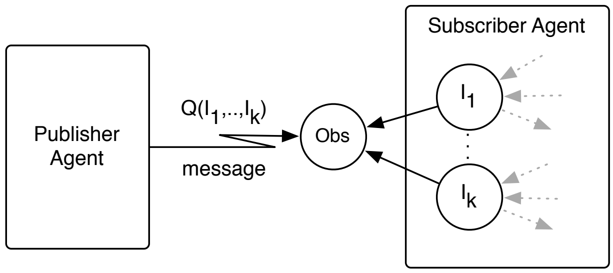


FIGURE 2. Introduction of an observation variable in a subscriber agent for absorption of a publisher message over shared variables $I_1, ..., I_k$.

To update an agent's distribution $P(V)$ with new evidence $Q(E_1, E_2, ..., E_n)$ for some set of observation variables $\{E_1, E_2, ..., E_n\} = I$ one calculates the joint probability $P(V)$, dividing by the marginal probability $P(I)$, and multiplying it by the new distribution of $\{E_1, E_2, ..., E_n\}$, this corresponds to the application of Jeffrey's rule,

$$Q(I) = Q(E_1) \cdot Q(E_2) \cdot ... \cdot Q(E_n), \tag{1}$$

thus obtaining:

$$Q(V) = P(V \backslash I|I) \cdot Q(I) = \frac{P(V)}{P(I)} \cdot Q(I). \tag{2}$$

In the case in which the input variables are not independent in the receiving agent, Equation 1 does not hold. (See (Valtorta et al., 2002, Section 5) for a detailed discussion on this point.) Lemma 1 in (Valtorta et al., 2002) allows the replacement of Equation 2 by:

$$Q^*(V) = P(V \backslash I|I) \cdot Q(I) = \frac{P(V)}{P(I)} \cdot Q^*_I(I), \tag{3}$$

where Q^*_I is the I_1-projection of probability distribution P on the set of all distributions defined on I and having $Q(E_i), i = 1, ..., n$, as their marginals. In practice, $P(V)$ could be updated to $Q^*(V)$ using the *big clique algorithm* of (Valtorta et al., 2002; Kim et al., 2004), *lazy big clique algorithm* of (Langevin and Valtorta, 2008), or the *wrapper methods* of (Pan et al., 2006).

Thus a mechanism similar to that already used for updating probabilities in a Bayesian network adjusts the world view of the agent, $P(V)$, into a conditional probability table $P(O|I)$. Note that this table is calculated using the local observations of the agent: $P(O|I) = \sum_L P(O, I, L)/P(I)$. It then combines that table with the external view of the inputs, $Q(I)$, to allow the calculation of the new values for the output variables $Q(O)$.

Given this view of the purpose of each agent in the overall system, an agent system may be considered an expansion of the Bayesian network formalism to a DAG where the distribution of the variables of one agent is obtained by conditioning on its input variables. This is not strictly the case for two reasons. First, when input variables are not independent in the receiving agent, then the calibration equation 2 must be replaced by the formally identical, but substantially and computationally more complex equation 3.

Second, the oracular assumption imposes the additional constraint that, in the agent system, unlike a Bayesian network, all parents are not affected by their descendants. More precisely, the only variables that may affect the variables in an agent are (1) those in the agent itself and (2) those in a preceding agent. In order to provide a formal definition of "preceding agent," we introduce the notion of communication graph in Section 2.2.3.

2.2.3. *Communication Graphs.* In order to represent the message passing and updating implications of AEBN's, we define a graphical representation of the agent system, called a *communication graph*. This graph is a DAG whose nodes are the agents and where edges are drawn from a publisher of shared variables to each of the subscribers of the shared variables. These edges are in turn labeled with the variables that they share. It is permissible for an agent to subscribe to only a subset of the published variables of another agent. In this case, the publishing agent will marginalize $Q(O)$ to the desired subset and pass this marginal to the subscriber agent.

We can now formalize the constraint that, in the agent system, all variables that are parents are not affected by their descendants. Let A_i and A_j be two distinct agents, let V_i, V_j be the sets of variables in agent A_i and A_j, respectively, and let $W_i \subseteq V_i$, $W_j \subseteq V_j$. Then if there is no directed path in the communication graph from A_j to A_i, any changes (whether by observation or by intervention) in the state of the variables in W_j does not affect the state of the variables in W_i. This is a very strong condition on the distribution of the variables in different agents of the agent system. This is *not* a symmetric relation, and therefore cannot be represented by any independence relation, since every independence relation is symmetric. There is an analogy to be made with casual Bayesian networks (Pearl, 2000). In a causal Bayesian network, when a variable is set (by external intervention), the parents of that variable are disconnected from it; more precisely, the result of the intervention is to create a new Bayesian network in which we remove the edges incoming into a variable that is set. The analogy, however, is not complete. In a causal Bayesian network, when a variable is set by intervention, some of the parent variables may be affected through backdoor paths, as explained in (Pearl, 2000, section 3.3). In an AEBN, there is no possibility for a variable in an agent to be affected by a descendent agent.

Consider as an example a four-agent system, where a supervisor agent fuses reports from two observer agents, each of which reports information from a single sensor agent. The communication graph shown in Figure 3 is constructed by first identifying shared variables (S, L_1, and L_2), then directing labeled edges from the producing agents to the consuming agents. The labels for the edges correspond to the shared variables. In this example, the edges directed from the *Sensor* agent to the $Observer_1$ and $Observer_2$ agents are labeled with S, and the edges from $Observer_1$ and $Observer_2$ to the *Supervisor* agent are labeled with L_1 and L_2, respectively. Henceforth this example will be referred to as the Redundantly Observed Sensor Example (ROSE).

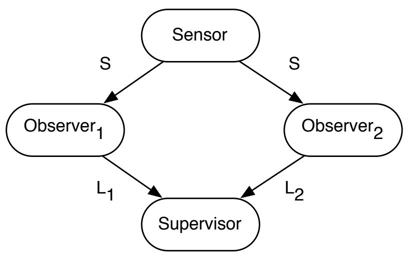


FIGURE 3. Redundantly Observed Sensor Example (ROSE) communication graph.

3. RUMOR PROBLEM

A central problem of message passing in probabilistic systems is the familiar rumor problem, where cycles in message passing cause redundant influence of beliefs. This problem is often known as the "rumor problem." In this chapter, we develop algorithms to identify and solve the rumor problem in the context of our multiagent system. Our identification algorithm adopts the approach proposed by Bloemeke (Bloemeke, 1998). In particular, we adopt the notion of a redundancy graph to identify the flow of rumors in the agent communication graph. To compensate for rumors, a communication solution is proposed that expands message passing. The solution highlights the challenges of correct message passing in probabilistic systems when rumors are present.

3.1. Redundant Influences

In decentralized systems, detecting and solving the rumor problem is critical to support normative decision making that is free of bias. Utete provided a useful "dining table" analogy (originally proposed by F. Banda) to highlight the effect rumors have in decentralized systems (Utete, 1998). Consider a dinner party where guests sit around a large oval table. The table is so large that guests can only communicate with their immediate neighbors. Guests can glean information from remote guests only through the communication of intermediaries. The host of the dinner party communicates a rumor to their neighbors, who in turn communicate the rumor to their neighbors. Eventually the information arrives from both sides of the table to the guest opposite of the host at the table. The dinner guest receives the same piece of information from two (assumed) independent sources, thus arriving at a biased view. If the guest were aware that both communications originated from the same source, the information may be handled differently. Figure 4 depicts the described analogy.

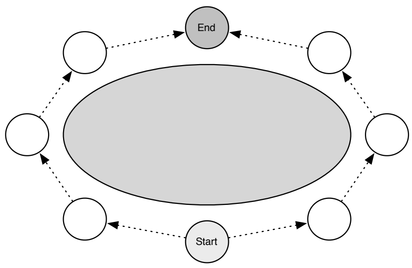


Figure 4. Example of a rumor initiated by the dinner party host, labeled "Start", spreading around a dining room table. The rumor reaches the guest labeled "End" from two distinct neighbors, doubly influencing them.

We now demonstrate the rumor problem in the context of our agent model using the ROSE communication graph as an example.

It is within the communication graph that the nature of the rumor problem can be clearly understood. Using the ROSE communication graph as an example, we can see the problem centers on the fact that the supervisor agent's view of the world, held in its Bayesian network, is doubly influenced by the initial sensor reading. The supervisor computes its belief in L as

$$P(L) = \sum_I P(L|I)P(I) \tag{4}$$

which expands to

$$P(L) = \sum_{L_1,L_2} P(L|L_1,L_2)\phi_{Ob_1}\phi_{Ob_2} \tag{5}$$

where ϕ_{Ob_1} and ϕ_{Ob_2} are messages the supervisor agent receives from $Observer_1$ and $Observer_2$. Expanding ϕ_{Ob_1} yields

$$\phi_{Ob_1} = \sum_S P(L_1|S)\phi_{Sensor} \tag{6}$$

with ϕ_{Ob_2} being calculated as

$$\phi_{Ob_2} = \sum_S P(L_2|S)\phi_{Sensor} \tag{7}$$

Finally, expansion of ϕ_{Sensor} yields

$$\phi_{Sensor} = P(S) \tag{8}$$

Substitution of equations 6, 7, and 8 into equation 5 leaves us with the following equation

$$P(L) = \sum_{L_1,L_2} P(L|L_1,L_2) \overbrace{\sum_S P(L_1|S)P(S)}^{Message\ from\ Observer_1} \overbrace{\sum_S P(L_2|S)P(S)}^{Message\ from\ Observer_2} \tag{9}$$

Finally, pulling the sums out leaves the following equation for $P(L)$

$$P(L) = \sum_{L_1,L_2,S} P(L|L_1,L_2)P(L_1|S)P(S)P(L_2|S)P(S) \tag{10}$$

In equation 10, $P(S)$ is redundantly incorporated in the supervisor agent, resulting in a redundantly influenced calculation of $P(L)$, because the correct expression of $P(L)$ (calculated using the chain rule) is

$$P(L) = \sum_{L_1,L_2} P(L|L_1,L_2)P(L_1,L_2) \tag{11}$$

Since there exists no directed path between L_1 and L_2 in the communication graph, they are independent by the oracular assumption and therefore:

$$P(L_1,L_2) = \sum_S P(L_1|S)P(L_2|S)P(S) \tag{12}$$

Substitution of equation 12 into equation 11 leaves the correct equation for calculating $P(L)$

$$P(L) = \sum_{L_1,L_2,S} P(L|L_1,L_2)P(L_1|S)P(L_2|S)P(S) \tag{13}$$

where equation 13 is the desirable outcome of message passing in the agent system and equation 10 is the actual outcome. Further, this problem can be made arbitrarily worse simply by adding additional paths between the sensor and the supervisor agents.

Redundant influences arise in a communication graph whenever the combination of messages received by an agent causes the belief in some shared variable to be over included.

A principal objective of this research is to allow for the handling of the rumor problem in an automated fashion. This will be achieved using algorithms that first identify redundant influences using the communication graph (Section 3.2) and then using a communication based solution (Section 3.3) to eliminate them.

3.2. Identifying Redundant Influences

This section describes a method for identifying redundant influences in a communication graph. V_i *redundantly influences* V_j if there exists multiple node-disjoint paths from V_i to V_j. There is a *redundant influence* between nodes V_i and V_j in some communication graph G if either V_i redundantly influences V_j or V_j redundantly influences V_i

Redundant influences are external to an agent and are dependent on the communication graph topology, therefore, it cannot be assumed they will be known in advance when the agent's internal model is designed. Hence, the redundant influences are orthogonal to the dependencies that are encoded in the Bayesian network that is contained within an agent and it is appropriate to support the processing of redundant influences separately from the construction of individual agents.

THEOREM 3.1 (Redundant influences occur on node disjoint paths): (Bloemeke, 1998) Given an AEBN communication graph $G = (V, E)$ with nodes $V_i, V_j \in V$, where V_i is an ancestor of V_j, it is sufficient to identify all node-disjoint paths from V_i to V_j in order to see all routes of redundant influence from V_i to V_j.

Following is a skeleton of a proof of Theorem 3.1. First we characterize thoroughly the routes through which redundant influences arrive at a node in the communication graph.

Assume that we have a communication graph G, as defined above, such that between V_i and V_j there are only n node disjoint paths $\{V_i \to V_{11} \to ... \to V_{1k_1} \to V_j, ..., V_i \to V_{n1} \to ... \to V_{nk_n} \to V_j\}$, where k_i, $1 \leqslant i \leqslant n$ is the length of path i minus the endpoints (5). Further, assume we have more than n redundant influences. Clearly, since redundant influences must occur along some series of paths, there must be some paths that are not node-disjoint between V_i and V_j causing the additional redundancies. Each of these additional paths must take on one of four forms (assume $p, q \in [1, n]$):

(1) It starts at one node along node-disjoint path q and ends at a different node along node-disjoint path p. (Figure 6(a))
(2) It starts at one node along path q and ends at node V_j. (Figure 6(b))
(3) It starts at node V_i and ends at a node along node-disjoint path q. (Figure 6(c))
(4) It starts and ends along node-disjoint path q. (Figure 6(d))

In all of these cases, the additional redundant influences are due to a subgraph that does not include both V_i and V_j. The subgraph effectively amplifies the redundancy between V_i and V_j, but the cause is redundant influences in the subgraph caused by multiple node disjoint paths between two nodes in the subgraph. If we were to compensate for these additional redundancies in the subgraph then we are left with the redundant influences from the n node disjoint paths between V_i and V_j. In other words, we can recursively remove redundant influences in subgraphs between V_i and V_j and be left with n redundant influences corresponding to the n node disjoint paths. The remaining n redundancies could be compensated for similarly. This is a recursive argument and means we can identify all redundant influences by examining all pairwise node disjoint paths in the graph.

The *Create Redundancy Graph* algorithm, described below, can be used to detect and label node disjoint paths in a communication graph. Once this algorithm has been run, a new graph, known as the *redundancy graph*, is constructed. The redundancy graph has the same nodes and edges as the communication graph, but its edge labels are expanded if and only if there are redundant influences. This graph will be used, along with the original communication graph, to compensate for redundant influences in the communication solution described in Section 3.3.

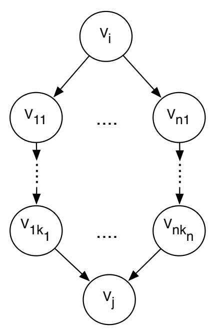


Figure 5. n node disjoint paths between V_i and V_j.

Algorithm 3.1 (Create Redundancy Graph Algorithm): Let $G = (V, E)$ be a communication graph and R be a copy of G that will serve as the redundancy graph. The edge labels of R will be expanded as described below.

(1) Let G' be a copy of G for use in a maximum-flow problem (Cormen et al., 2003).
(2) Modify G' by replacing each node v that has multiple incoming edges with two nodes v_1, v_2. Replace

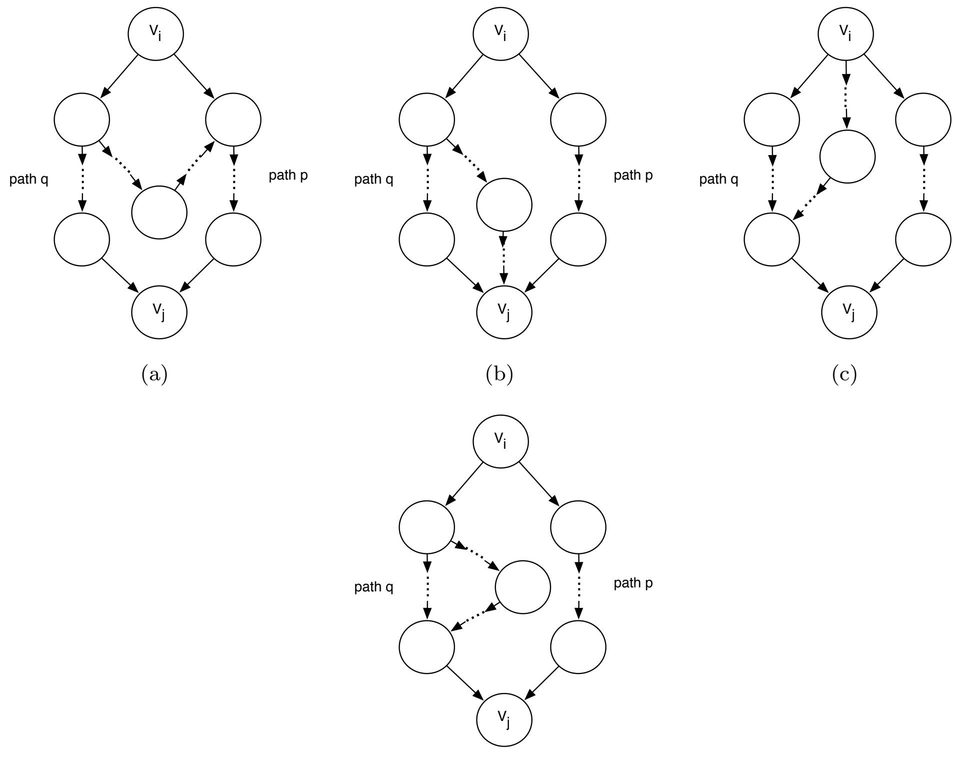


FIGURE 6. Extra redundant influences that are not node-disjoint (four cases).

each incoming edge to v, $< x, v >$, with a new edge $< x, v_1 >$, and each outgoing edge from v, $< v, y >$, with a new edge $< v_2, y >$. Finally, create a directed edge $< v_1, v_2 >$[4].

(3) For each pair of nodes v_s and v_t in G take each variable s_i that is produced by v_s:
   (a) Designate v_s as the source for the flow problem and v_t as the sink for the flow problem in G'.
   (b) Set the maximum flow of each edge in G' to 1.
   (c) Set the maximum flow to 0 for all outgoing edges in G' from v_s that have a label that does not contain variable s_i in G.
   (d) Run the maximum-flow problem on G'.
   (e) If the flow into the sink is greater than 1, then redundant influences exist between the two nodes:
      (1) For each edge in the maximum-flow problem solution that has flow greater than zero (and therefore is on a node disjoint path), add the shared variable s_i to the label of the corresponding edge in the redundancy graph R.
(4) Return the redundancy graph R.

It is well known that the maximum-flow problem can be used to find all node disjoint paths between a pair of nodes v_s and v_t (Kozen, 1992, Chap. 16). Hence the above algorithm will expand the edge labels of all node disjoint paths that a shared variable's influence travels for each combination of v_s, v_t. From Theorem 3.1, these node disjoint paths are also the redundant influences and therefore the algorithm will identify all redundant influences and expand the labels of the edges each redundant influence travels.

The time complexity of the algorithm to solve the maximum-flow problem described in (Kozen,

[4]Without this step, the algorithm would find edge-disjoint paths instead of node-disjoint paths.

1992) is $O(nm\log(\frac{n^2}{m}))$ where $n = |V|$ and $m = |E|$. Since a maximum-flow problem is executed for each pair of nodes, the total time for the *Create Redundancy Graph* algorithm is $O(n^3 m\log(\frac{n^2}{m}))$. This time is within $O(n^5)$ because m in the worst case is n^2. We note that since this is a distributed system, algorithms exist to find the maximum flow in $O(n^2\log^3 n)$ time using $O(n^2(\log^3 n + \sqrt{m}))$ communication complexity (Motyckova, 1995) but requires each node having knowledge of the communication graph (which can be accumulated in no worse than $O(m)$ time).

It is important to note that this is the only step in an AEBN system that requires global knowledge of the network topology. After the edges have been labeled no further knowledge outside of the immediate neighborhood is necessary.

Considering the ROSE example of Figure 3, the *Create Redundancy Graph* algorithm produces the redundancy graph of Figure 7. In this graph, only two edges have expanded labels. This arises because only in the case of the cycle (i.e., v_s = Sensor, v_t = Supervisor) will the flow problem return a flow greater than 1. In this case, all four of the edges between the Sensor node and the Supervisor node will have a flow of 1 and therefore will have S added to their edge label.

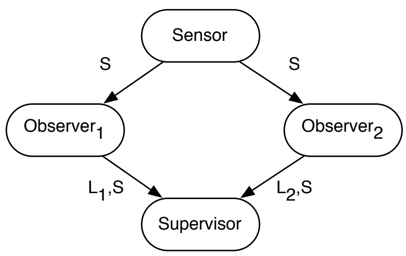


Figure 7. The ROSE redundancy graph.

3.3. Communication Solution

This section proposes a method of compensating for redundant influences where agent communication has been expanded to pass joint probabilities along the appropriately labeled links in the redundancy graph, without any change in the local Bayesian networks of each agent. Computing the expanded messages requires the probability update algorithm used by the agents to be flexible enough to allow the calculation of joint probabilities involving fixed input and output variables.

The calculation of joint probabilities is not trivial, especially in the presence of soft evidence. The soft evidential update algorithms such as the Big Clique (Valtorta et al., 2002) and Lazy Big Clique (Langevin and Valtorta, 2008) are based on the junction tree method, and were designed to compute all single variable marginals, but they can be used to compute one or several joint probabilities using techniques such as value or variable propagation, described in (Jensen, 1995, Section 5.1). Additionally, the wrapper methods (Pan et al., 2006) can be used similarly with a junction tree algorithm, or with a direct query-based algorithm such as bucket elimination.

Given the labeling in the redundancy graph, the correct (i.e., not redundantly influenced) probabilities can be retrieved at each agent without any modification in the Bayesian network it maintains, so long as the message that travels each edge is the joint probability of its label variables. Since no local Bayesian network model modification is necessary, this is known as the communication solution.

Care must be taken when removing redundant influences to ensure all redundant influences are correctly compensated for. This is done by ordering the removal of the redundant influences. Consider an example where an agent a_i receives three messages from neighboring agents: $\phi_1(A,B,C)$, $\phi_2(A,B,D)$, and $\phi_3(A,E)$. Agent a_i subscribes to the input variables C, D and E, hence its local Bayesian network calculates $P(O|C,D,E)$ and therefore needs to calculate the joint probability, $Q(C,D,E)$, from the received messages. The three messages contain redundant influences: $\phi_1(A,B,C)$ and $\phi_2(A,B,D)$ have redundant influence $\{A,B\}$, while $\phi_3(A,E)$, $\phi_1(A,B,C)$, and

$\phi_2(A, B, D)$ have redundant influence $\{A\}$ (8). If we first remove the redundant influence common to all, $\{A\}$, we can divide two of the messages by $Q(A)$ and have:

$$Q(B, C|A) = \frac{\phi_1(A, B, C)}{\sum\limits_{B,C} \phi_1(A, B, C)} = \frac{Q(A, B, C)}{Q(A)}$$

$$Q(B, D|A) = \frac{\phi_2(A, B, D)}{\sum\limits_{B,D} \phi_2(A, B, D)} = \frac{Q(A, B, D)}{Q(A)}$$

Next, we eliminate the remaining redundant influence, $\{B\}$, that is common to $Q(B, C|A)$ and $Q(B, D|A)$, by dividing one of the updated messages by $Q(B)$:

$$Q(C|A, B) = \frac{Q(B, C|A)}{\sum\limits_{A,C} \phi_1(A, B, C)} = \frac{Q(B, C|A)}{Q(B)} = \frac{Q(A, B, C)}{Q(A)Q(B)}$$

The correct joint probability is retrieved by combining the updated messages, $Q(C|A, B)$, $Q(B, D|A)$, and $Q(A, E)$,

$$Q(A, B, C, D, E) = Q(C|A, B)Q(B, D|A)Q(A, E)$$

Finally, the required joint probability of the inputs is calculated by marginalizing,

$$Q(C, D, E) = \sum_{A,B} Q(A, B, C, D, E)$$

This is correct, if $Q(A, B) = Q(A)Q(B)$, which means A and B are independent. However, this is not correct in general and therefore removal of the redundant influences in this order is incorrect. The removal of redundant influences must be ordered so as to not lose any dependence relations among the redundant influences. In this example, this can be achieved by first removing the redundant influence $\{A, B\}$ from $\phi_1(A, B, C)$, and finally removing the remaining redundant influence $\{A\}$ from $\phi_3(A, E)$. This can be restated, more generally, as ordering the received messages and removing the largest (in cardinality) remaining redundant influences. We now present formal definitions for this procedure, and then an efficient algorithm is described.

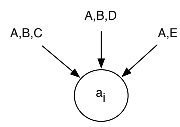


FIGURE 8. Multiple overlapping redundant influences.

Definition 4: Each *message* ϕ_i received by an agent a_k in the communication graph is a joint probability distribution over the variables that make up the corresponding label in the redundancy graph. Let $dom(\phi_i)$ be the domain of message ϕ_i.

Definition 5: A pair of messages ϕ_i and ϕ_j received by agent a_k contains *potential redundant information* if and only if $dom(\phi_i) \cap dom(\phi_j) \neq \emptyset$. We call $dom(\phi_i) \cap dom(\phi_j)$ the *redundant influence* or *redundant information* between ϕ_i and ϕ_j.

Since each shared variable has a unique label, and agent messages are expanded to pass shared variables that cause redundant influences, if a pair of messages ϕ_i and ϕ_j contain redundant information, they will have one or more common shared variables, i.e. $dom(\phi_i) \cap dom(\phi_j) \neq \emptyset$.

Definition 6 (Redundancy-free update rule): A message ϕ_i is updated to remove redundant information by the following update procedure:

$$\phi_i^* = \frac{\phi_i}{\sum_{D_i - D_i \cap D_V} \phi_i}$$

where $D_i = dom(\phi_i)$, $D_V = \bigcup_{j \neq i} dom(\phi_j)$ and ϕ_j has not been updated. If $D_i \cap D_V = \emptyset$, then the update is defined as the identity of ϕ_i.

The update rule states that removal of redundant information is achieved by dividing a message by the marginal of variables it has in common with other messages, excluding messages that have already been updated. It is necessary to exclude updated messages, because by definition they are no longer redundant with any other message. If the message does not share variables with any qualifying message, then the update does not change the message. Observe that if $D_i \cap D_V = dom(\phi_i)$, the resulting updated message is 1. Ex: $\frac{P(A)}{P(A)} = 1$. This can occur if the domain of a message is a subset of another message.

Definition 7: Let Φ be a set of messages received by agent a_k, ordered as $O = \phi_1, \phi_2, ..., \phi_n$, where n is the number of messages in Φ. We call the order O, the *update order*. If the redundancy-free update rule is applied to each $\phi \in \Phi$ according to the update order, then the resulting updated messages Φ^* are said to be *redundancy-free*.

We note that Φ^* does not depend on the update order, however, the order chosen will affect the efficiency of the update. We do not discuss update ordering considerations further in this research but note it is related to the variable elimination order problem.

In order to minimize the change of the receiving agents, independence relations in an agent's Bayesian network should be respected. This means we should not treat the joint probability over the input variables as one big joint distribution, otherwise we may make some variables independent after update that were not independent before update.

An agent a_k preprocesses its received messages by updating them to redundancy-free messages (Definition 7) which can be safely combined together to retrieve the uncontaminated joint over the input variables. This joint probability is marginalized to the required inputs which are joint distributions over dependent variables or single marginals (when no dependency exists). Since the calculation of a large joint distribution over all the inputs is often unnecessary and very expensive in both time and space, the following algorithm is a more efficient method that exploits the independence of the input variables and decomposes the calculation of the required inputs. The two methods result in the same calculated inputs:

Algorithm 3.2 (Remove Redundant Influences): Let Φ be a set of messages received by agent a_k.

(1) Partition Φ such that for each partition s_i, where $|s_i| > 1, \bigcap_{\phi \in s_i} dom(\phi) \neq \emptyset$.
(2) For each partition s_i where $|s_i| > 1$
   (a) Order the messages in s_i
   (b) Apply the redundancy-free update rule on the messages according to order

The messages can be processed in this manner due to Lemma 3.3 in Section 3.5, which proves independence of variables in messages that do not intersect. The result of the algorithm will be a set of messages that are redundancy-free. To absorb the messages into the agent's Bayesian network, the messages in each partition need to be combined. Each combination is added as a piece of soft evidence, as formalized in the algorithm below.

Algorithm 3.3 (Create Soft Evidence): Let S be the resulting partition of updated messages from the Remove Redundant Influences algorithm.

(1) For each partition $s_i \in S$
   (a) Combine the messages in s_i:

$$\phi_i = \prod_{\phi \in s_i} \phi$$

(b) Marginalize to needed inputs:

$$\phi_i = \sum_{dom(\phi_i)-I} \phi_i$$

Each ϕ_i is treated as soft evidence, and absorbed by creating observation variables and using soft evidential update.

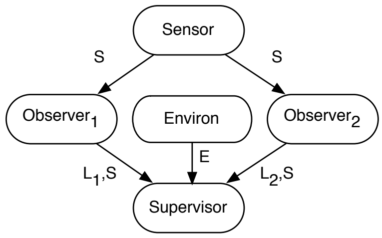


FIGURE 9. The extended ROSE redundancy graph.

Returning to an extended version of the ROSE example, where an additional agent responsible for monitoring the environment communicates its belief in the current environmental conditions to the Supervisor agent (Figure 9). Consider the problem of correcting the redundant influence on the Supervisor agent. The three incoming messages correspond to $m(Observer_1) = P(L_1, S)$, $m(Observer_2) = P(L_2, S)$ and $m(Environ) = P(E)$ which lead to the double counting of $P(S)$. To remove the redundant influence, the Supervisor agent invokes *Remove Redundant Influences*, which first partitions the received messages, which in this example results in two partitions:

$$s_1 = \{P(L_1, S), P(L_2, S)\}, s_2 = \{P(E)\}$$

Since $|s_1| > 1$, the messages are ordered as $< P(L_1, S), P(L_2, S) >$ and the redundancy-free update rule is applied on the messages according to the order, resulting in the updated messages:

$$s_1 = \{P(L_1|S), P(L_2, S)\}$$

The Supervisor agent invokes *Create Soft Evidence* to retrieve the appropriate soft evidence, which combines the updated messages in s_1 to give:

$$P(L_1, L_2, S) = P(L_1|S)P(L_2, S)$$

Which is marginalized to the required inputs:

$$P(L_1, L_2) = \sum_S P(L_1, L_2, S)$$

from which the desired probability $P(L)$ can be computed as:

$$\begin{aligned} P(L) &= \sum_{L_1, L_2, E} P(L|L_1, L_2, E) \sum_S P(L_1|S)P(L_2|S)P(S)P(E) \\ &= \sum_{L_1, L_2, E} P(L|L_1, L_2, E)P(L_1, L_2, E) \end{aligned}$$

where we note that $P(L_1, L_2, E)$ is calculated either using equation 2 or 3.

3.4. Coherent AEBN Systems

The semantics of AEBN require the marginal belief of shared variables be the same in each agent due to the oracular assumption. Bias in the system is avoided by detecting and correcting for the propagation of rumors. However, in particular agent communication topologies, the oracular assumption is violated when rumors are compensated for. We will explore this problem with an example.

Consider the following AEBN system composed of five agents with a communication graph as shown in Figure 10(a). It is clear to see that information from agents A and B doubly influence agent E through multiple node disjoint paths. To remove the rumors, the communication solution will expand the messages from agents C and D to pass joint distributions including α and β. The redundancy graph for this example is shown in Figure 10(b).

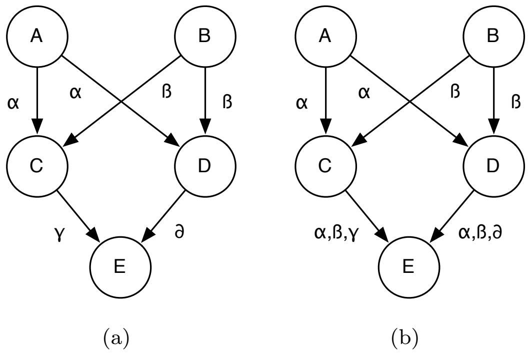


Figure 10. The communication graph (a) and resulting redundancy graph (b) illustrating the coherency problem.

In this example, agent A shares its marginal belief, $P(\alpha)$, with agents C and D. Likewise, agent B shares its marginal belief, $P(\beta)$, with agents C and D. Both agents C and D revise their internal models to be consistent with the beliefs received by treating the beliefs as soft evidence. Although $P(\alpha)$ and $P(\beta)$ will be respected in both agents, the joint probability $P(\alpha, \beta)$ is determined separately by each agent. In other words, the dependency between the variables α and β are modeled separately in agents C and D. The only restriction on $P(\alpha, \beta)$ in each agent is that the marginals $P(\alpha)$ and $P(\beta)$ respect the evidence received.

According to the communication solution, to correct for the rumors in agent E, both agents B and C must pass a joint probability that includes α, β. Agent E compensates for the rumor by conditioning one of the received messages by the shared redundant information, $P(\alpha, \beta)$.

No matter which message agent E conditions the oracular assumption will be violated. In either case, one of the marginals will not be the same as in the publishing agent.

Consider agent E chooses to condition $P(\alpha, \beta, \gamma)$ received from agent C. The redundancy-free joint probability over $\alpha, \beta, \gamma, \delta$ is then calculated as:

$$P(\alpha, \beta, \gamma, \delta) = \frac{P(\alpha, \beta, \gamma)}{\sum_\gamma P(\alpha, \beta, \gamma)} P(\alpha, \beta, \delta) \tag{14}$$

The oracular assumption states that the belief $P(\alpha, \beta, \gamma)$ must be the same in subscribing agent E as the publishing agent C. Since agent E will revise its beliefs to be consistent with $P(\alpha, \beta, \gamma, \delta)$, its belief $P(\alpha, \beta, \gamma)$ can be calculated as:

$$\begin{aligned}
P(\alpha, \beta, \gamma) &= \sum_\delta P(\alpha, \beta, \gamma, \delta) && (15)\\
&= \sum_\delta \frac{P(\alpha, \beta, \gamma)}{\sum_\gamma P(\alpha, \beta, \gamma)} P(\alpha, \beta, \delta) && (16)\\
&= \frac{P(\alpha, \beta, \gamma)}{\sum_\gamma P(\alpha, \beta, \gamma)} \sum_\delta P(\alpha, \beta, \delta) && (17)
\end{aligned}$$

However, there is no guarantee that $\sum_{\gamma} P(\alpha, \beta, \gamma) = \sum_{\delta} P(\alpha, \beta, \delta)$ since both are independently calculated in separate agents. The equality may not hold for several reasons:

(1) Agents C and D have differing conditional probability tables.
(2) Agents C and D have differing conditional dependence models.
(3) Agents C and D have differing d-separation induced by local evidence received by each agent.

Therefore, agent E may not respect the evidence received from agent C, violating the oracular assumption. It is trivial to see that the oracular assumption is violated even if agent E conditions $P(\alpha, \beta, \delta)$ instead of the first message.

Rumors can only be corrected for and global consistency maintained, if the AEBN system is *coherent*, i.e., no pair of agents have conflicting dependency relations amongst shared variables. We distinguish between two different types of coherence, *strong coherence* and *weak coherence*, in the following definitions.

Definition 8 (Strong Coherence): Two AEBN's, $A = (I_A, L_A, O_A, E_A, P_A), B = (I_B, L_B, O_B, E_B, P_B)$ are said to be *locally coherent* if the belief on their shared variables, $S_A = I_A \cup O_A, S_B = I_B \cup O_B$ agree: $P_A(S_A \cap S_B) = P_B(S_A \cap S_B)$. If a pair of agents are not locally coherent, they are said to be *incoherent*. An AEBN system is said to be *strongly coherent* if all pairwise AEBNs in the system are locally coherent.

Definition 9 (Weak Coherence): An AEBN system is said to be *weakly coherent* if the only locally incoherent agents are agents that share no common descendent.

In our AEBN model, we only desire the system be weakly coherent. It is permitted for agents to be locally incoherent if the conflicting information does not reach a common subscriber agent, causing a conflict. From this point forward we will refer to weak coherence as global coherence, unless the type of coherence needs to be distinguished.

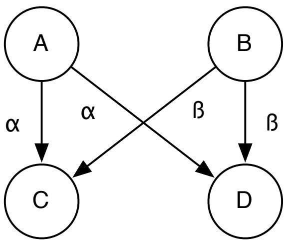


FIGURE 11. Example illustrating strongly and weakly coherent systems. If A, B, C and D are locally coherent then the system is strongly coherent. If only C and D are locally incoherent, then the system is weakly coherent.

We propose two different methods of ensuring global coherence in an AEBN system. The first method is a design time solution that imposes an additional constraint on an AEBN system. The first step of the design time solution is to identify potential sources of incoherence in the agent composition. The following algorithm analyzes the resulting redundancy graph of an AEBN system and marks sets of agents that are incoherent.

ALGORITHM 3.4 (Detect Incoherence): Let $\mathcal{A}$ be a set of agents in an AEBN system, C a communication graph for $\mathcal{A}$, and R the resulting redundancy graph.

(1) Let O be a topological ordering of the agents $\mathcal{A}$ in C
(2) Initialize $ANCESTORS(A) = \emptyset$ for each $A \in O$
(3) For each agent $A \in O$:

(a) Let Π be the set of parents of A in C

(b) Set $ANCESTORS(A) = \bigcup_{\pi \in \Pi}(ANCESTORS(\pi) \cup \{\pi\})$

(4) For each agent $A \in R$:

(a) For each pair of incoming edges to agent A, where I, the intersection of the edge labels has cardinality > 1:

(1) Let $\{X, Y\}$ be the set of parents of A that correspond to the pair of edges

(2) Let $\Pi = ANCESTORS(X) \cap ANCESTORS(Y)$

(3) If $\Pi \neq \emptyset$ and $\exists \pi \in \Pi$ with an outgoing edge labeled E, where $I \subseteq E$, then $\{X, Y\}$ are locally coherent

(4) Else, mark agents in Π that have no parents in R with an outgoing edge labelled E, where $I \subseteq E$

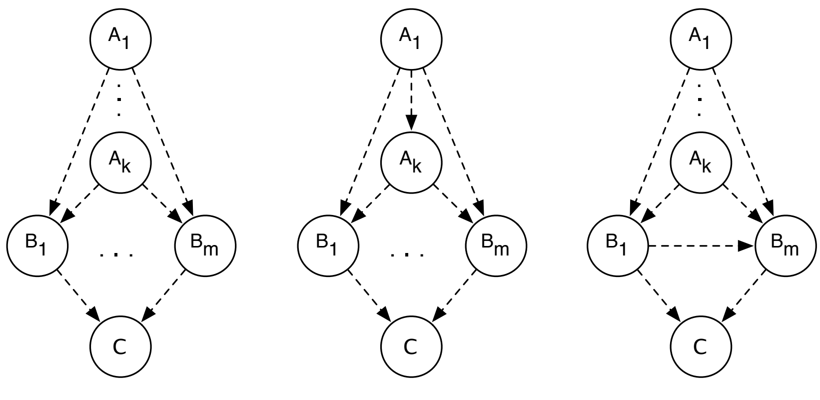
(5) If no agents have been marked, then the AEBN system is globally coherent

Algorithm 3.4 is used at design time to detect if the AEBN system is globally coherent, and if it is not, marks the nodes in the redundancy graph that are incoherent. The algorithm examines the redundancy graph and identifies multiple sources of a joint probability over shared variables. The worst case time complexity of this algorithm is $O(NE^2)$ where N is the number of agents, and E the number of edges in the communication graph. As discussed previously in this section, when multiple sources independently model dependency information over shared variables, the oracular assumption can be violated and prevent global consistency of the system. To prevent this from occurring, a new design constraint on AEBN systems is proposed, where only one agent is permitted to have oracular knowledge of the dependence relations amongst shared variables. This agent may not be the producer of the marginal beliefs of the shared variables, but is responsible for modeling the interactions amongst these shared variables, as is demonstrated in the running example in this section.

The coherence problem occurs when there exists two or more common ancestors between two or more agents. We formalize the coherence problem as follows:

Let $G = (V, E)$ be a DAG and let $C \in V$, $A = A_1, ..., A_{k-1}, A_k$, $B = B_1, ..., B_{m-1}, B_m$ where $k, m > 1, A \cap B \cap \{C\} = \emptyset$ and $A \cup B \in V$. Further, let each $B_i \in B$ be reachable from each $A_j \in A$ and C be reachable from each $B_i \in B$. Figure 12(a) shows the graphical structure defined. We say the graph is coherent if there exists a directed path that includes all nodes in A or a directed path that includes all nodes in B, otherwise the graph is incoherent.

As an example, Figure 12(b) shows a graphical structure where there exists a directed path in A, thus the graph is coherent.

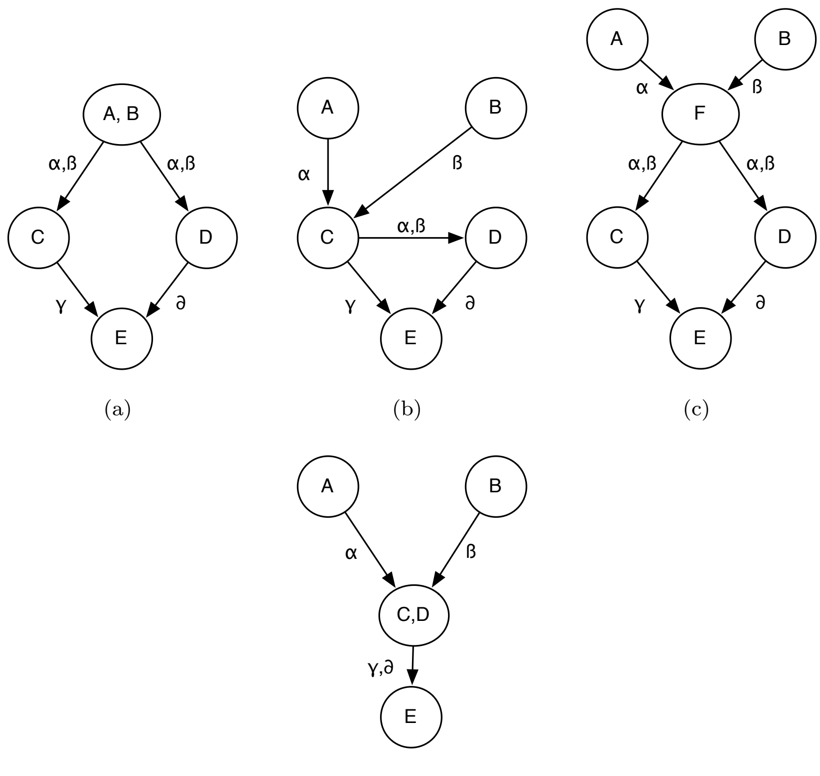


(a) (b) (c)

FIGURE 12. Graphical structure that leads to incoherence is shown in figure a), if there exists a directed path that includes all the nodes in A or all the nodes in B then the graph is coherent as shown in figure b) and c).

From the oracular assumption, we know that each agent's set of shared variables have unique labels but edge labels are expanded in the redundancy graph if multiple node disjoint paths exists between any two agents. Since an AEBN communication graph is a DAG, there exist no directed cycles. Therefore, if there exists two or more common ancestors between two or more agents, and the system is coherent, then there must exist a directed path between the common ancestors or between the agents (as shown in Figure 12(b) and 12(c)). This means one of the common ancestors must have an expanded label on an outgoing edge that contains the intersection. This is in fact what Algorithm 3.4 checks for and proves its sufficiency in determining coherence.

Running algorithm 3.4 on the redundancy graph in Figure 10(b), identifies that agents C and D are incoherent on $P(\alpha, \beta)$. To ensure global coherence of the system, we need to alter the agent composition so only one agent is responsible for determining $P(\alpha, \beta)$. Figure 13(a)-13(d) identify four possible design solutions that will result in a globally coherent system. In Figure 13(a), agents A and B are merged into one agent that is responsible for $P(\alpha, \beta)$ and shares this information with agents C and D. Figure 13(b) designates agent C as being responsible for $P(\alpha, \beta)$ and communication is altered so agent C passes this information directly to agent D, making communication from agents A and B to D unnecessary. Figure 13(c) introduces a new agent in the system that is responsible for $P(\alpha, \beta)$ and passing this information on to agents C and D. Finally, in Figure 13(d) agents C and D are merged into a single agent which results in a tree topology for the communication graph and hence the rumor problem no longer exist. Re-running algorithm 3.4 on each of these design solutions[5] returns they are globally coherent and while each has its own set of trade-offs, they all are sufficient to ensure global consistency of the system.



(a) (b) (c)

(d)

FIGURE 13. Design solutions for example in Figure 10(a) that ensure global coherence.

[5]There are many different possible design choices that could be made to ensure global coherence. The chosen solutions are for illustrative purposes and not an exhaustive set.

The second method proposed is a runtime solution. If restrictions of the problem domain prevent coherence being achieved at design time, then the AEBN communication graph can be modified to fuse the conflicting evidence. The fusion is performed by special "proxy" agents that are responsible for reconciling conflicting distributions at runtime. The reconciled distribution is then absorbed by the appropriate incoherent and subscribing agents. Figure 14 shows a proxy agent introduced to reconcile two incoherent agents. In this example, agents C and D are incoherent on $P(\alpha, \beta)$. The proxy agent F is introduced into the communication graph and communication links are updated so agent C and D no longer communicate directly with agent E, but rather, agent F acts as a proxy and communicates the redundancy-free belief of $P(\gamma, \delta)$ to agent E. This is necessary in order for agent F to first remove the incoherence by fusing the beliefs of agents C and D. After agent F has fused the beliefs received, it communicates the fused belief $P(\alpha, \beta)$ to agents C and D. To maintain consistency throughout the network it is necessary for F to send messages back to agents C and D. This appears to violate the AEBN assumption that an agent's belief cannot be affected by a subscribing agent. A proxy agent is a special type of agent that is permitted to violate this assumption in order to maintain global coherence in the system. After agents C and D receive the fused belief, they will revise their internal models to be consistent with this belief and recompute their messages retransmitting them to agent F. After this step, the agents C and D are locally coherent. Agent F can now safely remove any redundant influences between the received messages and calculate the message to send to agent E.

The following algorithm describes the process of adding necessary proxy agents to ensure runtime global coherence.

Algorithm 3.5 (Revise Communication Graph): Let $\mathcal{A}$ be a set of agents in an AEBN system, C a communication graph for $\mathcal{A}$, and R the resulting redundancy graph.

(1) Let $\mathcal{I}$ be the resulting incoherent agents determined by executing algorithm 3.4
(2) For each set $\mathcal{I}_i$ of locally incoherent agents in $\mathcal{I}$:
  (a) Instantiate a new proxy agent P and add to R
  (b) Associate a Fusion function with P
  (c) Revise the communication links in R to flow through proxy agent P:
    (1) For each agent A in $\mathcal{I}_i$:
      (A) Let $B = B_1, ..., B_{k-1}, B_k$ be the set of agents A communicates with.
      (B) Revise directed edges of the form (A, B_i) in R to (A, P) if edge label contains incoherent information
      (C) Create directed edges (P, B_i) and add variables in label associated with (A, B_i) to edge label. If a variable is already in edge label, then remove it from the label

Algorithm 3.5 instantiates a proxy agent and inserts it between sets of publishing agents that are locally incoherent and their subscribing agents. The communication edges are updated to route communication from the publishing agents to the subscribing agents through the proxy agent. Rumors are also removed by a proxy agent before sending messages on to subscribing agents, therefore it is only necessary for the communication edges between the proxy agent and subscribing agents to be labeled as the union of the edge labels associated with the publishing agents in the communication graph.

The task of the proxy agent is then to fuse incoherent messages received from the publishing agents and to remove rumors before passing the necessary information on to the subscribing agents on behalf of the publishing agents.

There are several different methods a proxy agent can use to fuse incoherent beliefs. The resulting belief can be calculated by the following different *Fusion functions*:

(1) Calculating a weighted average of the distributions, where the weight represents a trust (or reliability) score of the publishing agents. The scores could be maintained through experience, or fixed by a designer. If all agents are trusted equally then the result is an average of the distributions

(2) Designating one of the distributions as the true value, where the designation could be chosen randomly or through a trust mechanism, etc.
(3) Modeling the fusion as a disagreeing experts problem (Jensen, 2001, section 3.3.5).

After agents receive the fused beliefs, they revise their internal models to be consistent with the fused belief. This effectively removes any incoherence, resulting in a globally coherent system.

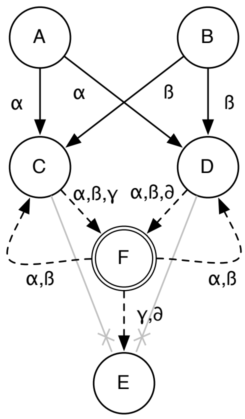


FIGURE 14. Runtime solution for example in Figure 10(a). Proxy agent F is introduced to reconcile conflict between agents C and D over $P(\alpha, \beta)$. The grey arrows are communication links that are removed in the original communication graph, and dashed lines are new communication edges introduced.

3.5. Proof of Communication Solution

We now prove the correctness of the communication solution for removing redundant influences.

Lemma 3.2: The domain of a message ϕ is invariant when applying the redundancy-free update rule. The resulting updated message ϕ^* is a conditional probability table.

Proof. Each message ϕ is a joint probability table, $P(dom(\phi_i))$. If the $dom(\phi)$ does not intersect the domain of any other message that has not been updated, the update rule returns the identity, ϕ, which is a conditional probability table with an empty set of conditionals: $P(dom(\phi)|\emptyset)$. Otherwise, by the definition of conditional probability,

$$P(X|Y) = \frac{P(X \cup Y)}{P(Y)}$$

which corresponds to the update rule, where $X \cup Y = dom(\phi), Y = dom(\phi) \cap \{\bigcup_j dom(\phi_j)\}$ where ϕ_j has not been updated. □

Lemma 3.3: Let Φ be a set of messages received by agent a_k. Let $V = \bigcup_{\phi \in \Phi} dom(\phi)$ be the variables of Φ. By the oracular assumption the variables in $dom(\phi)$ of message $\phi \in \Phi$, are independent of variables not in $dom(\phi)$, $V - dom(\phi)$.

Proof. We prove the lemma by contradiction:

Let $\phi_i, \phi_j \in \Phi$ be two messages received by agent a_k respectively from agents a_i and a_j (15). Assume there is a variable v_i only in $dom(\phi_i)$ and variable v_j only in $dom(\phi_j)$, and v_i is not independent of v_j.

From the oracular assumption, the beliefs of agent a_i are unaffected by its children in the communication graph. Therefore, the only agents that may influence a_i are its ancestors. From the stated assumption, v_i is not independent of v_j. There must be a directed path from the agent producing v_j to a_i.

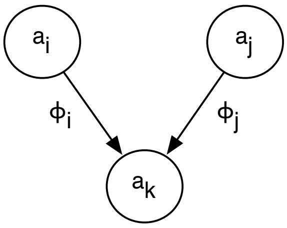


FIGURE 15. Messages received by agent a_k from agents a_i and a_j.

There are two cases:

(1) v_j is produced by a_j and a_j is an ancestor of a_i (Figure 16).
(2) v_j is produced by a common ancestor, a_s, of a_i and a_j (Figure 17).

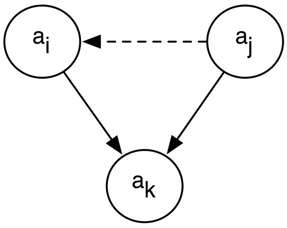


FIGURE 16. Agent a_j is an ancestor of agent a_i. The dotted line represents a directed path of any length (number of edges) greater than zero.

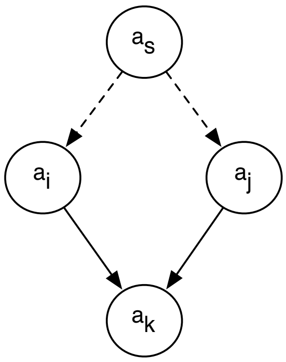


FIGURE 17. Agent a_i and a_j share a common ancestor a_s. The dotted lines represent a directed path of any length (number of edges) greater than zero.

In both cases, there exist multiple node disjoint paths between the agent producing v_j and a_k. The redundancy graph will extend the labels of the edges of these paths with v_j. In the communication solution, agent message passing is extended to pass messages over the labels in the redundancy graph. Since a_i is a node along one of these paths, the message ϕ_i sent to agent a_k must contain v_j. This contradicts the assumption and proves the lemma. □

Theorem 3.4: Let Φ be a set of messages received by agent a_k, and Φ has been updated to Φ^*, a set of redundancy-free messages. Let $V = \bigcup_{\phi \in \Phi^*} dom(\phi)$. For all $v \in V$, there is only one message

$\phi \in \Phi^*$ of the form: $P(X|Y)$, where $v \in X$, $X \subseteq V$, $Y \subseteq V - X$ and all other ϕ with $v \in dom(\phi)$ are of the form: $P(W|Z)$, where $v \in Z$, $Z \subseteq V$, $W \subseteq V - Z$. Less formally, this can be stated as: there exists only one message with v as a conditioned upon variable, and in all other messages with v in their domain, v is a conditioning variable.

Proof. Since each $\phi \in \Phi^*$ is redundancy-free, one of the following holds:

(1) v only belongs to the domain of one message, ϕ, and $dom(\phi)$ does not intersect with the domain of any other message. By Definition 6 and Lemma 3.2, the update of ϕ is the identity and therefore all variables in $dom(\phi)$ are on the left hand side of the conditioning bar. Since no other message shares variables with ϕ there exists only one message with v in its domain and it is of the form $P(X|Y)$, where $X = dom(\phi)$, and Y is the empty set.

(2) v only belongs to the domain of one message, ϕ, and $dom(\phi)$ intersects the domain of other messages. By Lemma 3.2, ϕ has been updated as:

$$\phi^* = P(X|Y)$$

where $X = dom(\phi) - Y$, $Y = dom(\phi) \cap \bigcup_{\phi_j \in \Phi^*, \phi_j \neq \phi} dom(\phi_j)$ and ϕ_j has not been updated. Since v only belongs to $dom(\phi)$, ϕ cannot be conditioned on v, and hence v is on the left hand side of the conditioning bar. Therefore there exists only one message with v in its domain and it is of the form $P(X|Y)$, where $v \in X$.

(3) v belongs to the domain of more than one message. By Definition 7, each message $\phi_i \in \Phi^*$ with $v \in dom(\phi_i)$ will be updated in some order $< \phi_{i_0}, ..., \phi_{i_k} >$ where k is the number of messages. Updating conditions each message on the intersection of its domain with all other messages that have not yet been updated. Since v belongs to the domain of all ϕ_i, all but the last message updated in the order will be conditioned on v. The last message ϕ_{i_k} either will intersect other messages that have not yet been updated and do not have v in their domain, or will have an empty intersection (and hence return identity). In either case, v will exist in only one message with the form $P(X|Y)$ where $v \in X$, and all other messages with v in their domain will have the form, $P(W|Z)$, where $v \in Z$.

From the three cases above, it follows that each $v \in V$ is uniquely a conditioned upon variable in one message, and a conditioning variable in all other messages that contain v in their domain. □

Theorem 3.5: If Φ^* is a set of redundancy-free messages received by agent a_k, then the joint probability $P(V)$, where $V = \bigcup_{\phi \in \Phi^*} dom(\phi)$ is:

$$P(V) = \prod_{\phi \in \Phi^*} \phi$$

Proof. We will show the combination of the messages in Φ^* results in the joint probability over V, $P(V)$.

Order the messages in Φ^* according to the reverse order redundancy-free update was invoked on the messages. We call this the *message order*. Let $\phi_1, \phi_2, ..., \phi_n$ be the message order, where n is the number of messages in Φ^*.

According to the redundancy-free update rule, each message is conditioned on the variables it shares with other messages that are later in the update order. Since the messages have been ordered the reverse of the update order, we can rewrite the update as:

$$\phi_i^* = \frac{\phi_i}{\sum_{dom(\phi_i) - dom(\phi_i) \cap \bigcup_{j=1}^{i-1} dom(\phi_j)} \phi_i}$$

Therefore, by Theorem 3.4, each message has a set of unique variables that are conditioned upon, $R_i = dom(\phi_i) - \{dom(\phi_i) \cap \bigcup_{j=1}^{i-1} dom(\phi_j)\}$. We call R_i the *residual set* of an updated message. Let $S_i = dom(\phi_i) \cap \bigcup_{j=1}^{i-1} dom(\phi_j)$, be the conditioning variables for message ϕ_i. From Lemma 3.2, the updated message is the conditional probability table:

$$\phi_i^* = P(R_i|S_i)$$

Each R_i is a disjoint subset of V, such that $\bigcup_i R_i = V$. Order the residual sets according to the message order. We call this the *residual sets order*. Let $R_1, R_2, ..., R_n$ be the resulting residual sets order, where $R_1 \subseteq dom(\phi_1), R_2 \subseteq dom(\phi_2), ..., R_n \subseteq dom(\phi_n)$.

We need to show that,

$$P(V) = P(R_n, R_{n-1}, ..., R_1) = \phi_n \phi_{n-1} ... \phi_1$$

We prove this by using induction on the number of messages in Φ^*. For n=1 we have,

$$P(V) = P(R_1) = \phi_1$$

since there are no other messages intersecting $dom(\phi_1)$, $R_1 = dom(\phi_1)$ and the updated message $\phi_1 = P(dom(\phi_1))$.

We will assume the theorem holds for any residual sets order created from $n = i$ messages,

$$P(V) = P(R_i, R_{i-1}, ..., R_1) = \phi_i \phi_{i-1} ... \phi_1$$

We will prove for $n = i + 1$,

$$P(V) = P(R_{i+1}, R_i, R_{i-1}, ..., R_1) = \phi_{i+1} \phi_i \phi_{i-1} ... \phi_1$$

By the definition of conditional probability we have,

$$P(R_{i+1}, R_i, ..., R_1) = P(R_{i+1}|R_i, R_{i-1}, ..., R_1) P(R_i, R_{i-1}, ..., R_1)$$

Since $R_{i+1} \subseteq dom(\phi_{i+1})$, and from Lemma 3.3, the variables not in $dom(\phi_{i+1})$ are independent of the variables in $dom(\phi_{i+1})$,

$$\begin{aligned}
P(R_{i+1}|R_i, R_{i-1}, ..., R_1) &= P(R_{i+1}|dom(\phi_{i+1}) - R_{i+1}) && (18)\\
&= P(R_{i+1}|dom(\phi_{i+1}) \cap \bigcup_{j=1}^{i} dom(\phi_j)) && (19)\\
&= P(R_{i+1}|dom(\phi_{i+1}) \cap \{R_i, R_{i-1}, ..., R_1\}) && (20)\\
&= P(R_{i+1}|S_{i+1}) && (21)\\
&= \phi_{i+1} && (22)
\end{aligned}$$

From Equation 22 and the inductive hypothesis,

$$\begin{aligned}
P(V) = P(R_{i+1}, R_i, ..., R_1) &= P(R_{i+1}|R_i, R_{i-1}, ..., R_1) P(R_i, R_{i-1}, ..., R_1)\\
&= \phi_{i+1} P(R_i, R_{i-1}, ..., R_1)\\
&= \phi_{i+1} \phi_i \phi_{i-1} ... \phi_1
\end{aligned}$$

□

We have therefore shown that using the communication solution, each agent can retrieve the correct distribution over the received shared variables, and marginalize this distribution to the required input distributions. Using equations 2 or 3, the agent updates its local Bayesian network to be consistent with the received messages.

4. EVALUATION

To evaluate the proposed AEBN model, we implemented an AEBN framework using the Java SE software development kit (JDK 1.6). This implementation allows us to run agent simulations and capture various performance metrics. These performance metrics are compared with similar simulations implemented using Xiang's MSBN framework and provide insight into the trade-offs of modeling using our AEBN model, and MSBNs. Performance metrics are collected for all agents during each phase of message passing in the agent communication graphs. In order to compare the two systems, we collect the following performance metrics:

(1) Cross-entropy and CD distance w.r.t. MSBN distribution of shared variables
(2) Posterior beliefs of shared variables
(3) Total size of all messages sent

The first two metrics provide insight into the effect the oracular assumption has on the shared beliefs in the agent system as compared with a system that strictly adheres to d-separation properties in a centralized graphical model. Since posterior beliefs in MSBN are identical to those in a global Bayesian network model, we will compare the belief of each shared variable in our agent model to the corresponding belief in a similar simulation implemented as an MSBN. The last metric provides insight into the computational and resource implications of our model and MSBNs.

Definition 10 (I-divergence): I-divergence (also known as Kullback-Leibler distance or cross-entropy) is a measure of the distance between two joint probability distributions $P(V)$ and $Q(V)$:

$$I(P||Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$$

Definition 11 (CD-distance): Chan and Darwiche proposed a new distance measure *CD-distance* (Chan and Darwiche, 2005; Darwiche, 2009) for measuring belief change. They note that even though I-divergence is a popular measurement of the distance between two joint probability distributions, it is not a true distance measure since it is not symmetric. Further, I-divergence only captures an average-case bound on belief changes.

The CD-distance measure is a true distance measure[6] that captures a worst-case bound on belief change, which Chan and Darwiche argue provides a more accurate measure of distance and is easier to calculate. CD-distance is calculated as:

$$CD(P, Q) = \log \max_x \frac{Q(x)}{P(x)} - \log \min_x \frac{Q(x)}{P(x)}$$

where $\frac{0}{0} = 1$ and $\frac{\infty}{\infty} = 1$.

Our desire is for the multiagent simulation to be semi-realistic to reflect design issues a multiagent system designer may face designing real world systems. Our chosen simulation is based on a "bio-attack" example devised by Laskey and Levitt (Laskey and Levitt, 2002). In this example, a sophisticated coordinated multi-city bio-warfare attack is orchestrated by a terrorist organization on the United States. The terrorist organization utilizes multiple contagions to masquerade a deadly anthrax attack as a less serious cutaneous anthrax and foot-and-mouth disease outbreak in the american cattle industry. All three contagions have similar symptoms in cattle and humans.

The goal of the terrorist organization is to cause government authorities to mistakenly link illness in humans from a deadly strain of anthrax with two independent disease outbreaks in cattle. The ensuing confusion will delay detection of the terrorist plot, resulting in high civilian casualties and high economic damage.

Although the example is fictitious, it is semi-realistic due to the following facts (Laskey and Levitt, 2002):

(1) Outbreaks of foot-and-mouth disease on livestock has the potential of causing trillion dollar economic damage to the US economy.

[6]A true distance measure must satisfy the three properties of distance: positivity, symmetry, and the triangle inequality.

(2) Over 95% of beef processing in the United states is concentrated in a very small number of large scale factories, mainly located in large industrial cities such as Chicago, Kansas City, Denver and Dallas/Fort Worth. The animal-to-product cycle is highly efficient and it takes only a few days for the product to reach the dinner table.
(3) Cutaneous anthrax can be transmitted to humans from livestock.
(4) Inhalation anthrax is deadly to both humans and livestock and is easily spread in aerosol form. Only 50-100kg of weapons grade anthrax would be required to attack an urban population.

The sequence of events for the scenario are outlined in Table 1. The scenario proceeds from day 1 (the start of the scenario) to day 18 (the end of the scenario) for a coordinated terrorist attack. We stop at day 18 since a terrorist attack is certainly detected due to the discovery of weapons grade inhalation anthrax in human populations. In the table, the events that are evidence the intelligence agents can gather are highlighted in bold. The goal of our simulation is to determine how well an AEBN and MSBN system can detect the terrorist attack. Note that the word agent in the phrase "terrorist agents" in Table 1 does not refer to an intelligent computational agent.

TABLE 1. Bio-attack sequence of events

Day	*Event*
Day 1	Terrorist agents infect Chicago cattle herds at target stockyards with cutaneous anthrax.
Day 3	Terrorist agents infect Chicago cattle herds with foot-and-mouth disease.
Day 5	**First reports of anthrax and foot-and-mouth symptoms in Chicago cattle herds.** Terrorist agents spray Chicago herds with inhalation anthrax. Simultaneously, terrorist agents infect Kansas City cattle herds at target stockyards with cutaneous anthrax.
Day 7	Terrorist agents use crop duster to spray Chicago with inhalation anthrax. Simultaneously, terrorist agents infect Kansas city cattle herds with foot-and-mouth disease.
Day 8	**Lab tests confirm cutaneous anthrax at Chicago stockyard.**
Day 9	Terrorist agents spray Kansas city cattle herds with inhalation anthrax. Simultaneously, terrorist agents infect Denver cattle herds at target stockyards with cutaneous anthrax.
Day 11	Terrorist agents use crop duster to spray Kansas city with inhalation anthrax.
Day 12	**Lab tests confirm inhalation anthrax at Chicago stockyard.** Terrorist agents infect Denver cattle herds with foot-and-mouth disease.
Day 13	**Lab tests confirm foot-and-mouth disease at Chicago stockyard.** Terrorist agents spray Denver herds with inhalation anthrax.
Day 14	**Lab tests confirm cutaneous anthrax at Kansas city stockyard.**
Day 15	**Lab tests confirm cutaneous anthrax at Denver stockyard.** Terrorist agents use crop duster to spray Denver with inhalation anthrax.
Day 16	**Lab tests confirm inhalation anthrax at Kansas city stockyard.**
Day 17	**Lab tests confirm inhalation anthrax at Denver city stockyard.**
Day 18	**Lab tests confirm inhalation anthrax in human populations in Chicago.**

To detect and reason about the scenario, we implement a fictitious distributed detection network that attempts to detect the unfolding scenario and minimize damage from the terrorist plot. Laskey and Levitt proposed a single Bayesian network[7] to illustrate the power of MEBNs, we constructed a similar Bayesian network (Figure 18) for reasoning about the scenario. We will use this model as a guide for constructing two multiagent systems: one based on our AEBN model, and the other based on an MSBN.

[7]Constructed using Multi-Entity Bayesian Networks (MEBN).

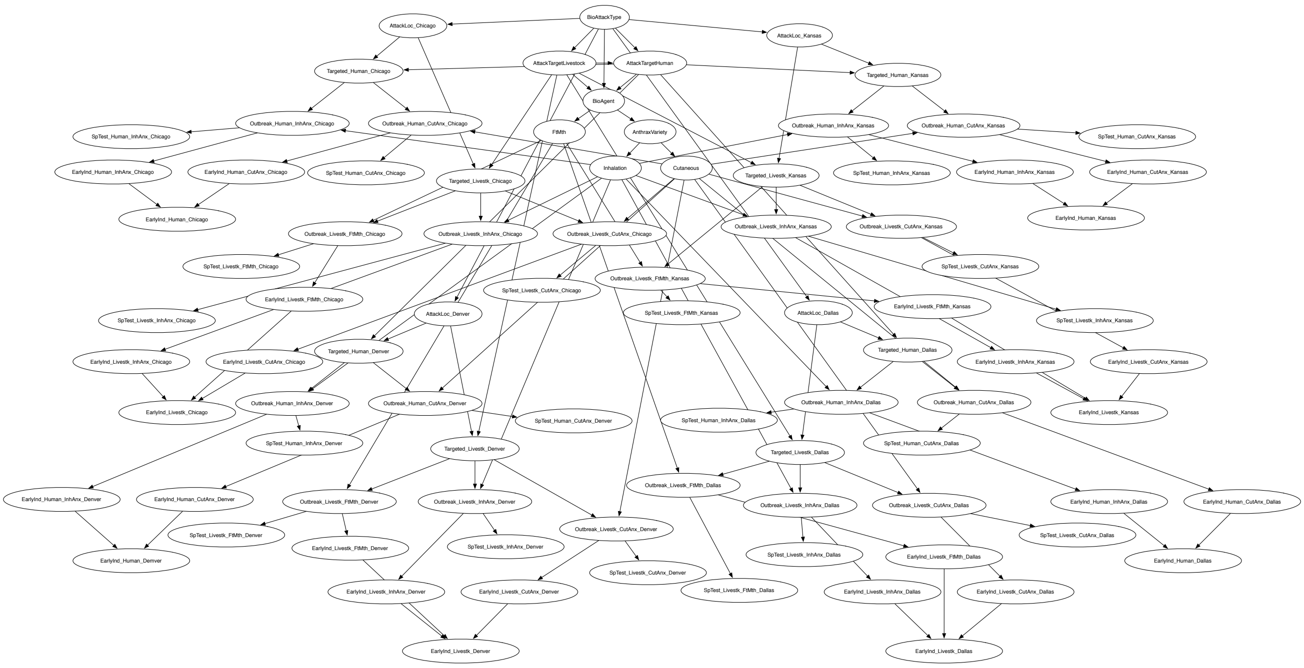


FIGURE 18. Bio-attack full Bayesian network.

Our simulation includes early detection agents representing agents located in meat processing facilities, and governmental monitoring agencies. Information gathered from early detection agents is reported to local threat assessment agents as part of a nationalized monitoring and detection network. Reports from local threat assessment agents are transmitted to a national incident agent which is responsible for assessing attack types detected and issue alerts to appropriate authorities of the probability of a coordinated terrorist attack. Figure 19 shows a summarized overview of the proposed agent communication graph for the AEBN simulation (only Chicago and Kansas agents are shown). The full multiagent system contains seventeen agents: one incident agent, eight attack type agents and eight early indicator agents. The attack type and early indicator agents are divided by region, where each region has two attack type agents and two early indicator agents for human and livestock population monitoring and testing. In our simulation, there are four regions: Chicago, Kansas, Denver and Dallas. The agent set associated with each region are essentially identical, except variable labels are specific to each particular region. A similar agent decomposition is devised for an MEBN multiagent system as discussed in Section 4.5.

4.1. Early Indicator Agents

Early indicator agents represent early detection report agents that monitor abnormal rates of illness or deaths in human and livestock populations and calculate their belief the observations indicate possible anthrax or foot-and-mouth disease. The joint probability of early indicators is calculated and passed to the appropriate attack type agent.

4.2. Attack Type Agents

Attack type agents represent early government monitoring and testing facilities that receive early detection reports from early indicator agents and also are capable of performing tests for specific strains of anthrax and foot-and-mouth disease. Each attack type agent computes its belief that an inhalation, cutaneous or foot-and-mouth disease outbreak in the target population has occurred given all the available evidence. The joint probability of the outbreak is calculated and passed to the incident agent.

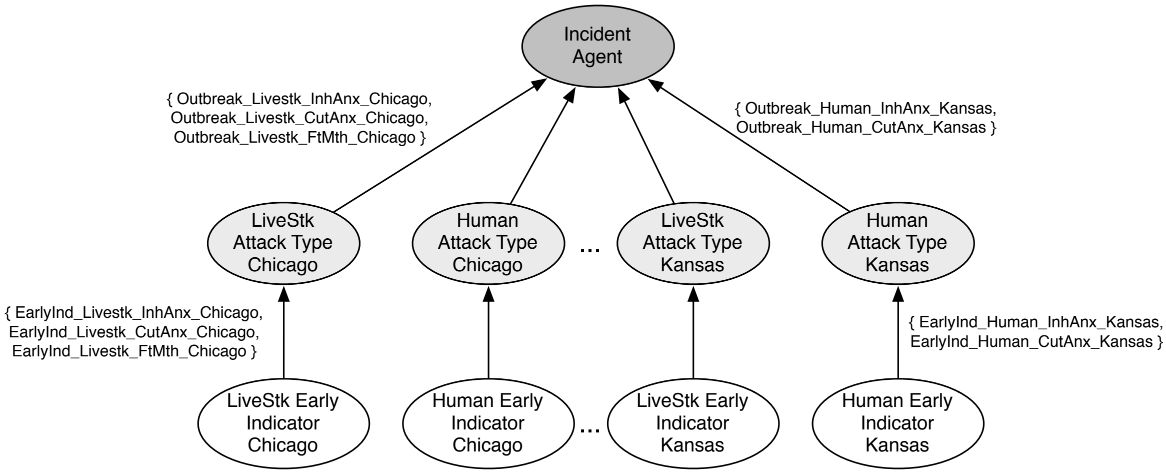


FIGURE 19. Communication graph for bio-attack AEBN simulation (only Chicago and Kansas agents shown).

4.3. Incident Agent

The incident agent represents a national alert agent that receives reports of outbreaks from attack type agents and assesses the probability of a terrorist attack and characterizes the type of terrorist attack. The incident agent relies on the reports from the attack type agents and fuses the information into its internal model. We envision in a real world scenario the incident agent would initiate alerts to appropriate authorities if the belief of a terrorist attack reached an appropriate threshold.

4.4. AEBN Multiagent Simulation

In our AEBN multiagent simulation, each agent calculates and processes messages according to the following:

(1) Messages are only sent to subscribers when new evidence is discovered
(2) Messages are computed using current beliefs based on all available evidence
(3) Replace previous evidence received with new evidence received
(4) The most recent evidence is used to reason

Each agent in an AEBN has an internal Bayesian network used for reasoning given local and external evidence received. The Bayesian network of the incident agent is shown in Figure 20. An example of the Bayesian networks for the attack type agents is shown in Figure 21 and Figure 23 for the Chicago human and livestock attack type agents respectively. Finally, an example of the Bayesian networks for the early indicator agents is shown in Figure 22 and Figure 24 for the Chicago human and livestock early indicator agents respectively.

In the Bayesian network figures, the shaded nodes with dashed borders represent observation nodes (as defined in Section 2.2) that are introduced to absorb the messages from publishing agents, and nodes that are shaded with double lined borders represent variables that can be subscribed to by other agents.

For the purposes of our simulation, we assume perfect communication [8] in the agent system. Each agent first receives all messages from publishing agents it is subscribed to then performs belief revision. After new beliefs have been computed, each agent computes and sends messages to subscribing agents. This process is performed for each evidence phase described in Section 4.6.

[8]Perfect communication means no latency, transmission failures or corrupted messages occur in the network.

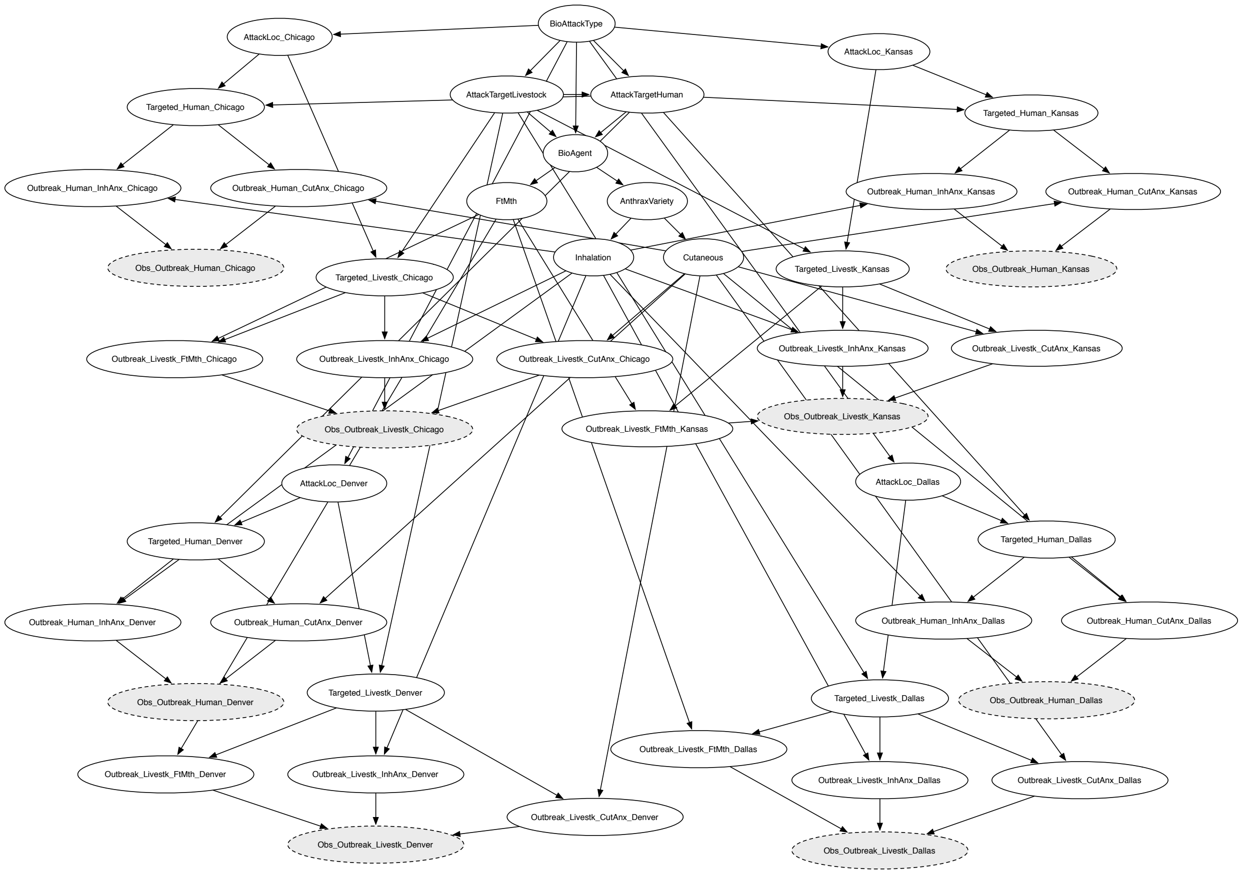


FIGURE 20. Bayesian network for Incident agent.

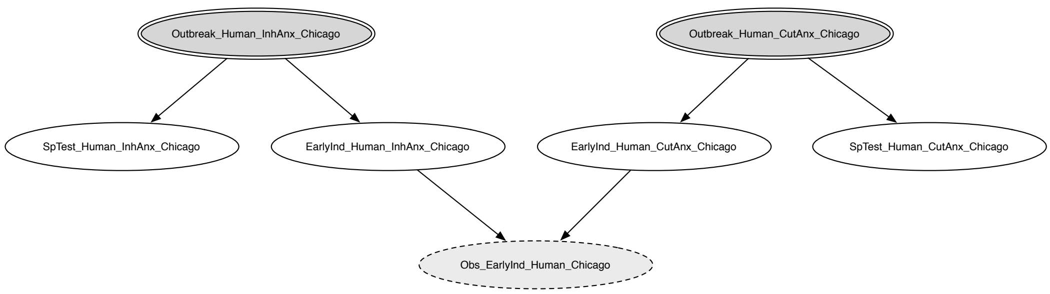


FIGURE 21. Bayesian network for Chicago Human Attack Type agent.

4.5. MSBN Multiagent Simulation

The MSBN multiagent simulation is constructed using a similar decomposition of the global Bayesian network as the AEBN simulation. We used Xiang's publicly available WEBWEAVER-III toolkit[9] to construct and validate the soundness of sectioning (see Section 2.1 for details) of the MSBN. Figure 37 shows a summarized version of the resulting linked junction forest for the MSBN. Each junction tree in the linked junction forest represents an agent. The agent roles are defined similarly to our AEBN decomposition and comprises the same set of seventeen agents.

[9]WEBWEAVER-III is available for download at http://www.cis.uoguelph.ca/∼yxiang/

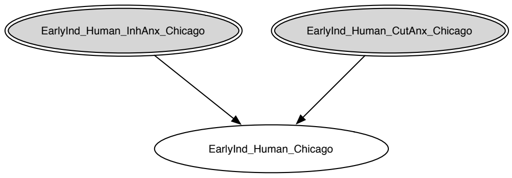


Figure 22. Bayesian network for Chicago Human Early Indicator agent.

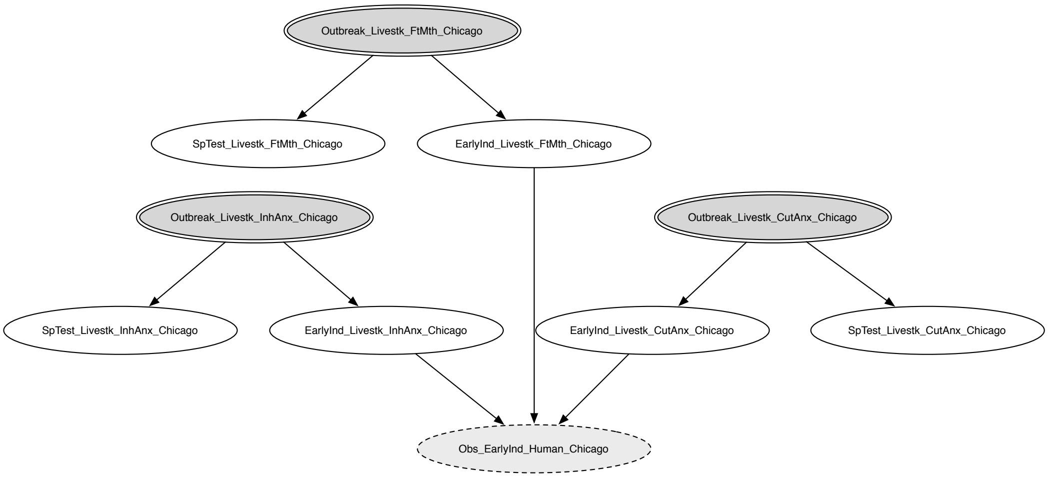


Figure 23. Bayesian network for Chicago Livestock Attack Type agent.

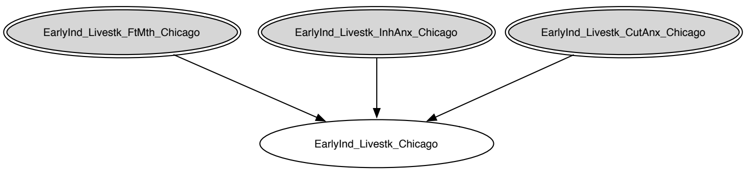


Figure 24. Bayesian network for Chicago Livestock Early Indicator agent.

Belief propagation in an MSBN is analogous to propagation in a junction tree where branches of the junction tree are distributed to agents, rather than centralized in one agent. Therefore, as opposed to our AEBN simulation, where agents only transmit messages when new evidence has arrived, in MSBN, messages are transmitted in both directions for each agent during each message passing phase similar to the collect and distribute evidence phases in a junction tree propagation algorithm. A full message passing phase is initiated when an agent receives new evidence and revises its beliefs. The evidence is propagated throughout the agent system so all agents are consistent over their shared beliefs given the new evidence.

In our MSBN simulation, each message passing phase corresponds to one agent receiving evidence. The evidence phases are described in Section 4.6. As with the AEBN simulation, we assume perfect communication in the MSBN simulation.

4.6. Evidence Phases

From the sequence of events defined in Table 1, the evidence phases defined in Table 2 will be used for the simulations. Each phase is defined as evidence for that phase being entered in the appropriate agent, which initiates message passing and belief update on the agent system according to the semantics of each simulation.

Since the goal of the simulations is the detection and classification of the terrorist attack we

TABLE 2. Evidence phases for Bio-attack simulation.

Phase 1	Initial state (no evidence)
Phase 2	Sick cows observed in Chicago
Phase 3	Positive test for cutaneous anthrax in Chicago livestock
Phase 4	Positive test for inhalation anthrax in Chicago livestock
Phase 5	Positive test for foot-and-mouth disease in Chicago livestock
Phase 6	Positive test for cutaneous anthrax in Kansas city livestock
Phase 7	Positive test for cutaneous anthrax in Denver livestock
Phase 8	Positive test for inhalation anthrax in Kansas city livestock
Phase 9	Positive test for inhalation anthrax in Denver livestock
Phase 10	Positive test for inhalation anthrax in Chicago human population

will focus on comparing each simulation's beliefs of the BioAttackType variable, which has states: Coordinated Bio Attack, Local Bio Attack, Non-Bio Attack, and No Attack.

4.7. Enhanced AEBN model

During the course of our experimentation, we discovered a modeling issue with our originally defined AEBN simulation. External evidence received by attack type agents had too strong an influence over local evidence. We identified this as a general modeling issue with AEBN systems and this behavior may not be appropriate in some situations. For example, when an external observer agent reports their unreliable belief of the presence of a disease to a subscriber agent which can obtain local evidence in the form of a test that can confirm or deny the existence of the disease with a high accuracy, the observation received is not as important as the local evidence.

In general, we can state the problem as: the reliability or importance of the external evidence needs to be modeled so it is offset or discounted by more reliable or important evidence the local agent acquires.

The situation in our simulation is similar, where our early indicator agents make observations that suggest (or indicate) the presence of anthrax or foot-and-mouth disease, and the attack type agents can perform a highly accurate test to confirm or deny the presence of the contagions.

To account for these type of situations we propose the following modeling technique in AEBN systems:

(1) Introduce a mediating variable in the subscriber agent
(2) The mediating variable acts as a "switch" that turns off or discounts the affect of the external evidence when more accurate local evidence is present

Figure 25 shows the general modeling technique, and the mediating variable conditional probability table is shown in Table 3. In our implementation, the external evidence is ignored (using a uniform distribution) if a test has been performed, but more generally it could be discounted using any suitable distribution. In practice, either the discount factor could be specified by a designer or an agent could maintain a discount factor based on the historic reliability of the communicating agent.

TABLE 3. Conditional probability table for discount variable.

Test	*false*		*true*	
Disease	*false*	*true*	*false*	*true*
false	1	0	0.5	0.5
true	0	1	0.5	0.5

In Figures 26 and 27 we show the revised Bayesian network models of the Chicago human

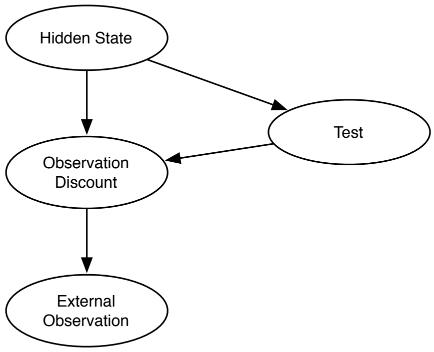


FIGURE 25. Discounting external evidence given an accurate test modeling technique.

and livestock attack type agents. The Kansas, Denver and Dallas attack type agents are modified similarly.

We will refer to this revised model as the Enhanced AEBN simulation or simply AEBN v2, and the first AEBN simulation as original AEBN simulation or AEBN v1.

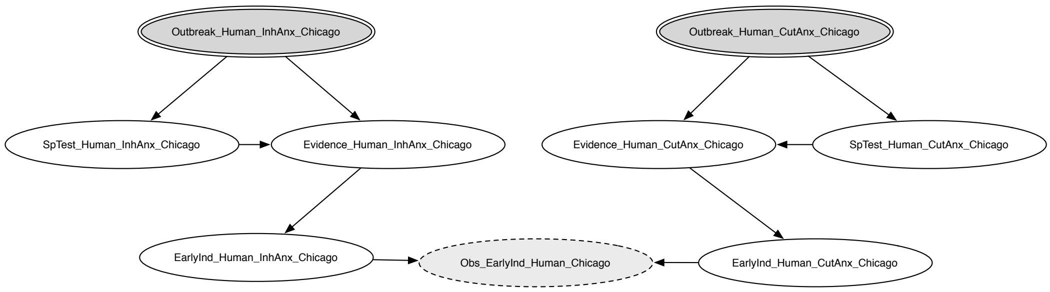


FIGURE 26. Revised Bayesian network for Chicago Human Attack type agent.

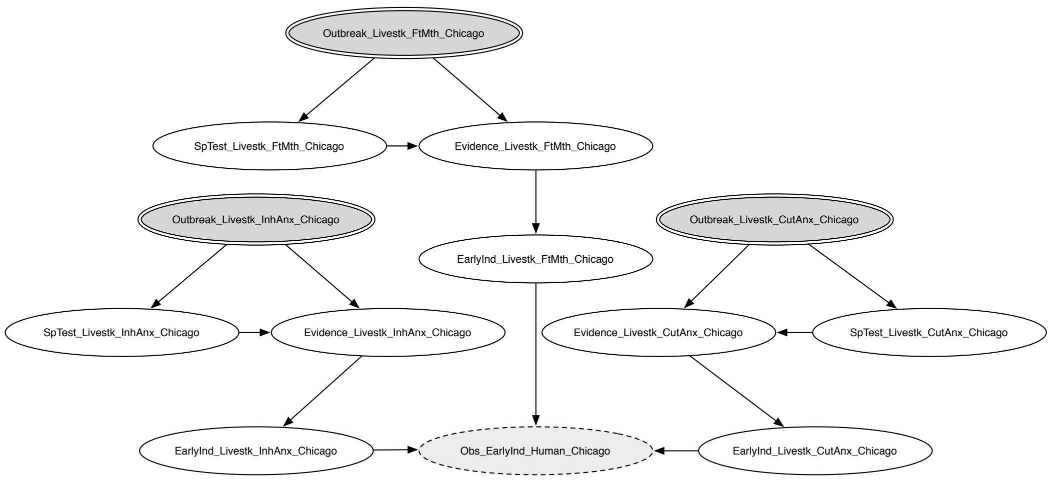


FIGURE 27. Revised Bayesian network for Chicago Livestock Attack type agent.

4.8. Simulation Results

In this section we present the results of our simulations. As discussed in the beginning of this chapter, several performance metrics were collected to compare the predictive ability and efficiency of the simulations.

Tables 4, 5, 6 and 7 show the belief of the states of the BioAttackType variable in the three simulations (MSBN, AEBN v1 and AEBN v2) as the evidence phases progress. Figures 28, 29, 30 and 31 show the corresponding plots. From these plots we see AEBN v2 performs closer to the MSBN simulation than AEBN v1. Both AEBN simulations respond similarly to the MSBN simulation, but are not as sensitive to the evidence presented in the evidence phases.

The MSBN detects the coordinated terrorist attack at phase 5 with a belief of 63.262%, while AEBN v1 detects the coordinated terrorist attack at phase 9 with a belief of 71.290% and AEBN v2 detects the coordinated terrorist attack one phase earlier at phase 8 with a belief of 68.463%. The superior performance of MSBN can be accounted for by the loss of some dependence relationships in the AEBN models due to the Oracular assumption. We draw the analogy of AEBN being like a naıve Bayes model and MSBN being like a general bayes model. However, AEBN is far more powerful than a naıve Bayes model since it does not sacrifice all dependence relationships.

We note that in a realistic setting, an elevated risk of a terrorist attack of even a modest amount would trigger a national terrorist alert. If we applied such an approach in the incident agent with an alert threshold of 10% an MSBN would detect an unfolding terrorist attack one phase earlier at phase 4 and our AEBN model at phase 5 for our enhanced model and at phase 6 for our original AEBN model. This earlier detection could mitigate some of the damage to civilians and livestock in Kansas and Denver.

TABLE 4. Beliefs of Coordinated Bio Attack over scenario phases.

	MSBN - CoordAttck	AEBN v1 - CoordAttck	AEBN v2 - CoordAttck
Phase 1	0.01	0.010	0.010
Phase 2	0.014	0.012	0.011
Phase 3	0.148	0.071	0.086
Phase 4	8.373	0.912	1.452
Phase 5	63.262	4.939	10.430
Phase 6	90.241	18.734	39.809
Phase 7	93.508	27.068	48.031
Phase 8	95.181	49.479	68.463
Phase 9	95.971	71.290	81.353
Phase 10	97.829	78.172	86.828

To determine a principled measure of the difference of the distributions of the BioAttackType variable of the simulations, we calculated the CD-distance and I-divergence[10] of MSBN and AEBN v2. Only AEBN v2 was compared to MSBN since the plots in Figure 28-31 indicate it is overall superior to AEBN v1. The resulting distance measure results are shown in Figure 32 plotted over the evidence phases.

The I-divergence is largest during phase 5, which matches our intuition by inspecting the plots of belief change that show the largest change in belief in the MSBN simulation for Coordinate Bio Attack as $\Delta 55$ and No Attack as $\Delta 61$, while in the AEBN v2 simulation the change is $\Delta 4$ and $\Delta 6.4$ respectively. Overall, the I-divergence between the two simulations is not very large with the highest value being 1.257 during phase 5 and all other phases being less than 1. This indicates that AEBN v2 performs closely to MSBN overall.

[10]Both CD-distance and I-divergence were calculated using lg rather than ln or log.

Table 5. Beliefs of Local Bio Attack over scenario phases.

	MSBN - LocalAttck	AEBN v1 - LocalAttck	AEBN v2 - LocalAttck
Phase 1	0.040	0.040	0.040
Phase 2	0.059	0.056	0.049
Phase 3	0.337	0.237	0.264
Phase 4	3.573	1.327	2.014
Phase 5	9.760	3.677	6.983
Phase 6	8.417	13.475	22.443
Phase 7	6.152	19.184	26.154
Phase 8	4.803	25.234	22.044
Phase 9	4.025	17.673	12.296
Phase 10	2.169	16.770	9.418

Table 6. Beliefs of Non-Bio Attack over scenario phases.

	MSBN - NonBioAttck	AEBN v1 - NonBioAttck	AEBN v2 - NonBioAttck
Phase 1	0.1500000	0.1500000	0.1500000
Phase 2	0.1500000	0.1502337	0.1501284
Phase 3	0.1510000	0.1515000	0.1515951
Phase 4	0.1380000	0.1527319	0.1540521
Phase 5	0.0620000	0.1661524	0.1773863
Phase 6	0.0040000	0.1156992	0.0745359
Phase 7	0.0010000	0.0910298	0.0499422
Phase 8	0.0008840	0.0436203	0.0169584
Phase 9	0.0004119	0.0186277	0.0110681
Phase 10	0.0002104	0.0142564	0.0098572

The CD-distance results differ dramatically from I-divergence and identify a weakness in using this measure as a distance between distributions. CD-distance captures the worst case distance between two distributions where as I-divergence is a weighted average. We feel the weighted average is more representative and explains why I-divergence is a popular metric for comparing two distributions. As discussed above, one would expect the distance to be greatest during phase 5, rather than phase 10 as indicated by CD-distance. In our comparison, CD-distance of the two distributions is a monotonically increasing function, which is counter intuitive. CD-distance does not weight the distance by the likelihood of events as does I-divergence which we feel is a strong weakness of this measure and limits its applicability for providing an accurate measure of the variability between two distributions. However, CD-distance has some nice properties such as being a true distance measure and bounding the difference of beliefs captured by two probability distributions. For our purposes, these properties are not needed since we are comparing distance of our AEBN simulation to MSBN which we treat as a "gold standard", hence symmetry of our distance measure is not needed, nor is bounding of belief difference.

Finally, Figure 33 shows the communication cost of AEBN and MSBN over the scenario phases. Over all evidence phases, the AEBN has lower communication cost due to messages only in one direction: from publisher to subscriber. Additionally, agents only send messages when they have

TABLE 7. Beliefs of No Attack over scenario phases.

	MSBN - NoAttck	AEBN v1 - NoAttck	AEBN v2 - NoAttck
Phase 1	99.8	99.800	99.800
Phase 2	99.777	99.782	99.789
Phase 3	99.364	99.540	99.498
Phase 4	87.917	97.608	96.380
Phase 5	26.916	91.218	82.409
Phase 6	1.338	67.676	37.673
Phase 7	0.339	53.657	25.765
Phase 8	0.016	25.244	9.475
Phase 9	0.004	11.019	6.341
Phase 10	0.001	5.044	3.744

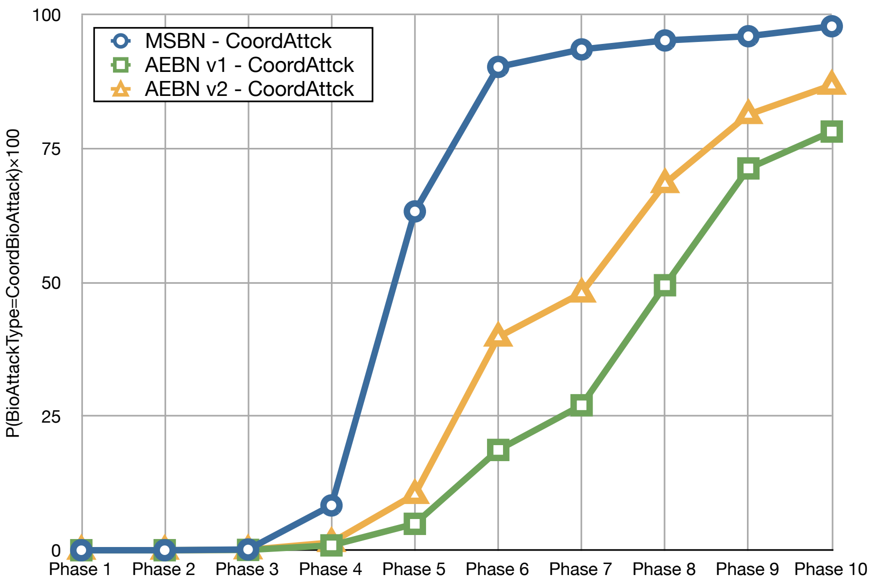


FIGURE 28. Beliefs of Coordinated Bio Attack over scenario phases.

revised their beliefs which can result in considerable savings in communication cost as can be seen in Phases 2 to 10. Conversely, the MSBN simulation communication cost is constant because each evidence phase corresponds to propagating evidence in both directions over the links in the link tree to maintain global consistency.

For large agent networks where evidence is seldom received by a subset of agents, we posit an AEBN system can have significantly better communication performance over an MSBN system, provided the network graph is sparsely connected. In the next section we analyze the scalability of AEBN systems in terms of the communication cost versus network size and number of network links.

4.9. Scalability of AEBN Multiagent Systems

To evaluate the scalability of AEBN systems we first perform worst case communication analysis. Consider an AEBN system of n agents, which are ordered and labelled according to each agent's

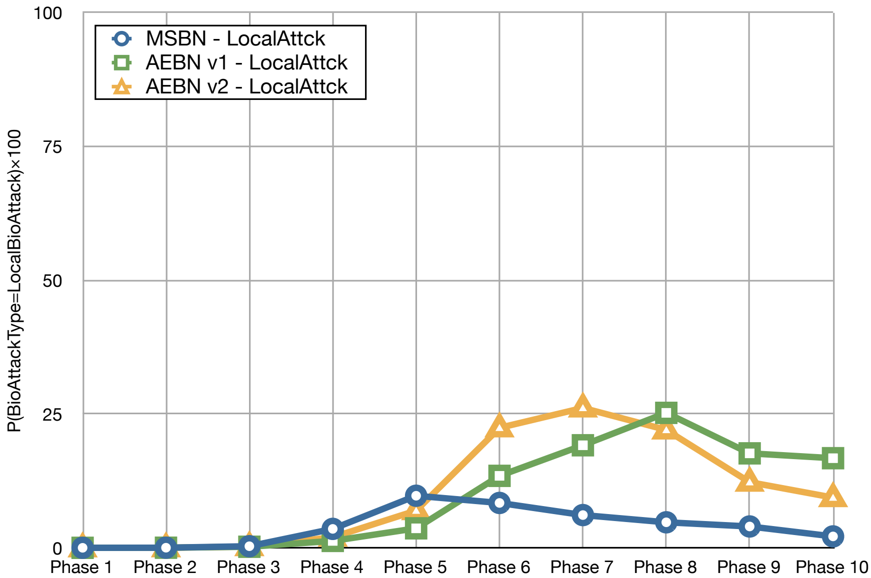


FIGURE 29. Beliefs of Local Bio Attack over scenario phases.

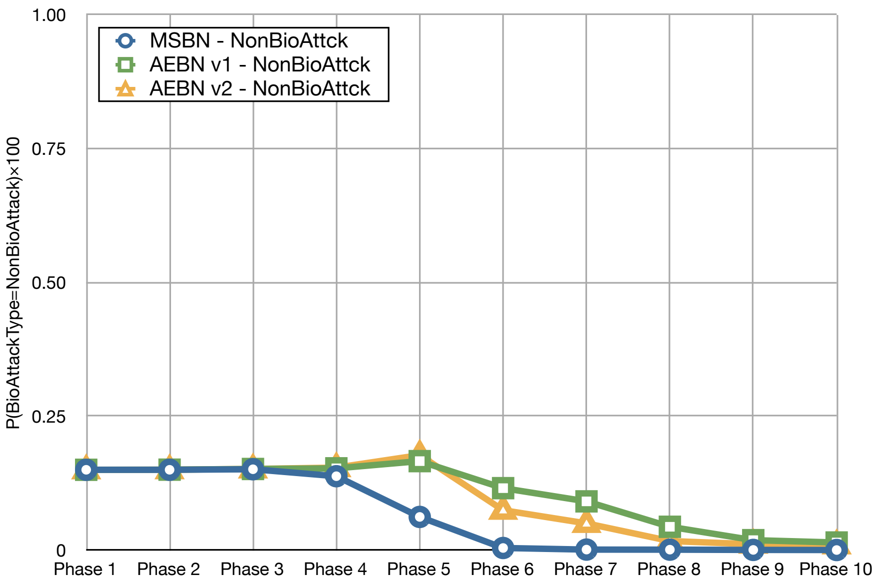


FIGURE 30. Beliefs of Non-Bio Attack over scenario phases.

index in the order. The worst case communication graph corresponds to a fully connected DAG. We create this communication graph as follows:

(1) Let G be a communication graph over n agents. Let O be an ordering of the agents, where i refers to the ith agent in O.
(2) For each agent i in the ordering O
  (a) Add a directed edge from i to all agents that follow i in the order O.

In this communication graph each agent will have $i - 1$ incoming edges and $n - i$ outgoing edges. For simplicity, let us assume each agent sends a message of size 2 along each of its outgoing

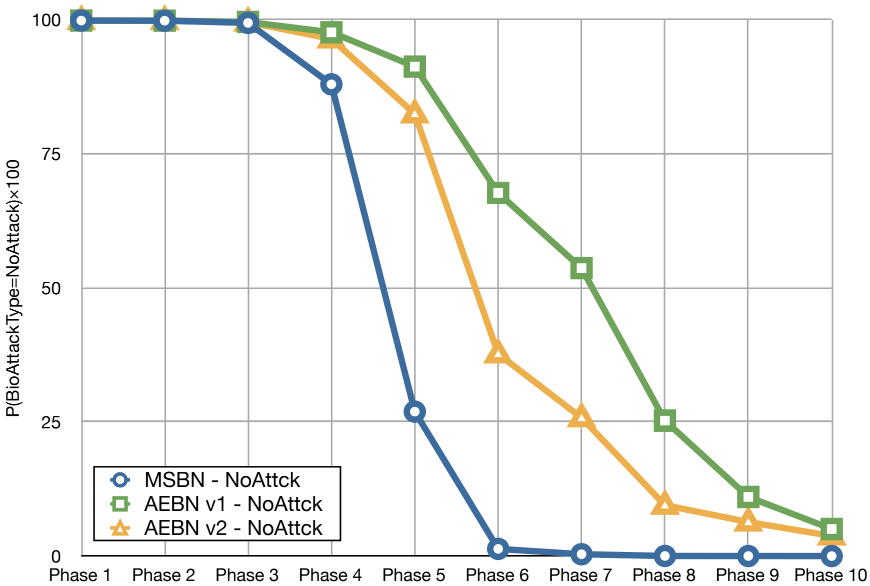


FIGURE 31. Beliefs of No Attack over scenario phases.

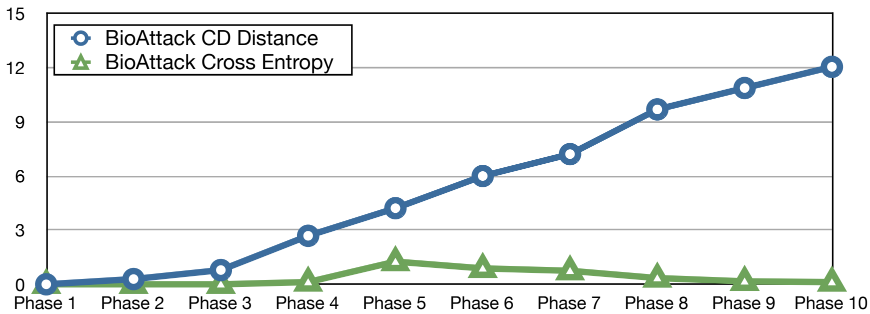


FIGURE 32. CD and I-divergence of BioAttackType distribution in AEBN v2 and MSBN over scenario phases.

edges (note the size of the message is not important since we are only concerned with an asymptotic upper bound on the communication complexity). Further, since multiply connected nodes exist in the graph, rumors are present in the communication graph. Each agent i has a directed edge from all agents with a label $< i$. Each of these agents, j, will carry redundant information from agents with a label $< j$. Therefore, any AEBN system with this communication graph will require the communication solution to avoid the influence of redundant information and expand the edge labels accordingly. Figure 34 depicts a graphical representation of the resulting redundancy graph.

The following summation calculates the cost of expanded communication in this redundancy graph:

$$\sum_{i=1}^{n-1}(n-i)*2^i$$

Which is bounded above by:

$$\sum_{i=1}^{n-1}(n-i)*2^i < n*n*2^n = O(n^2 2^n)$$

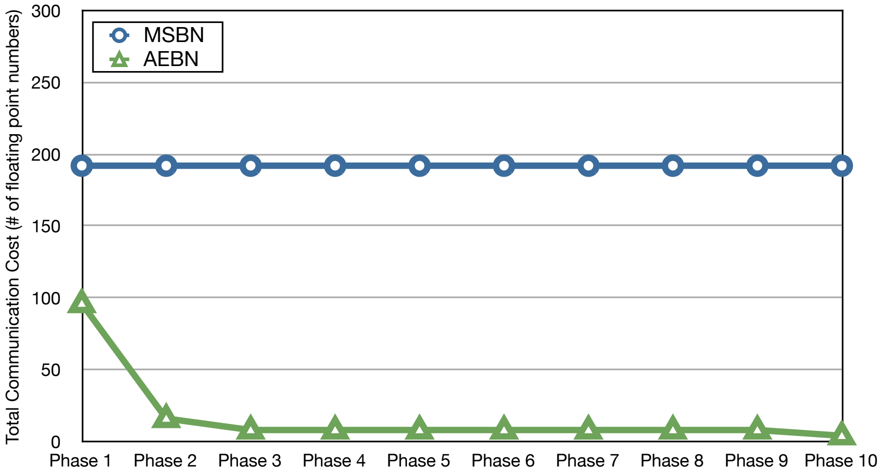


Figure 33. Total communication cost of AEBN v2 and MSBN over scenario phases.

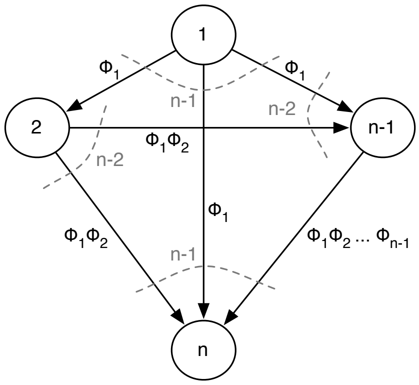


Figure 34. Theoretical scalability of AEBN.

This result shows the communication complexity in an AEBN system is intractable for densely connected communication graphs. If the communication is sparsely connected and undirected cycles are minimized, communication in AEBN systems is tractable which we evaluate experimentally.

To analyze the impact of the communication graph density on an AEBN system's communication complexity, we generated random communication graphs where we controlled the number of agents in the graph and number of edges. We first generated a random spanning tree over the agents to ensure the graph is not disconnected and randomly added additional edges, ensuring that the resulting graph was a DAG.

Let N be the number of agents in the graph, then the initial spanning tree has $N-1$ edges. This agent graph represents the minimum agent communication graph and is the lower bound on communication cost where communication cost is linear in the number of agents: $2 * (N-1)$. To explore the effect graph density has on the communication cost, we generated five sets of graphs where the number of edges were fixed at: $(N-1)$, N, $1.25N$, $1.5N$ and $2N$. For each set, we generated graphs with six different fixed number of agents: 5, 10, 20, 40, 80 and 160. Since there is much variability in the topology of the generated graphs, 20 graphs of each configuration were generated and average communication costs calculated. The results of the generated graphs is shown in Figure 35.

These results indicate that when the number of edges in the graph is $1.25 * N$ or below, the average cost of communication is linear or near linear in the number of agents, when the size of the network is below 160 agents. As the number of agents increases the cost grows exponentially.

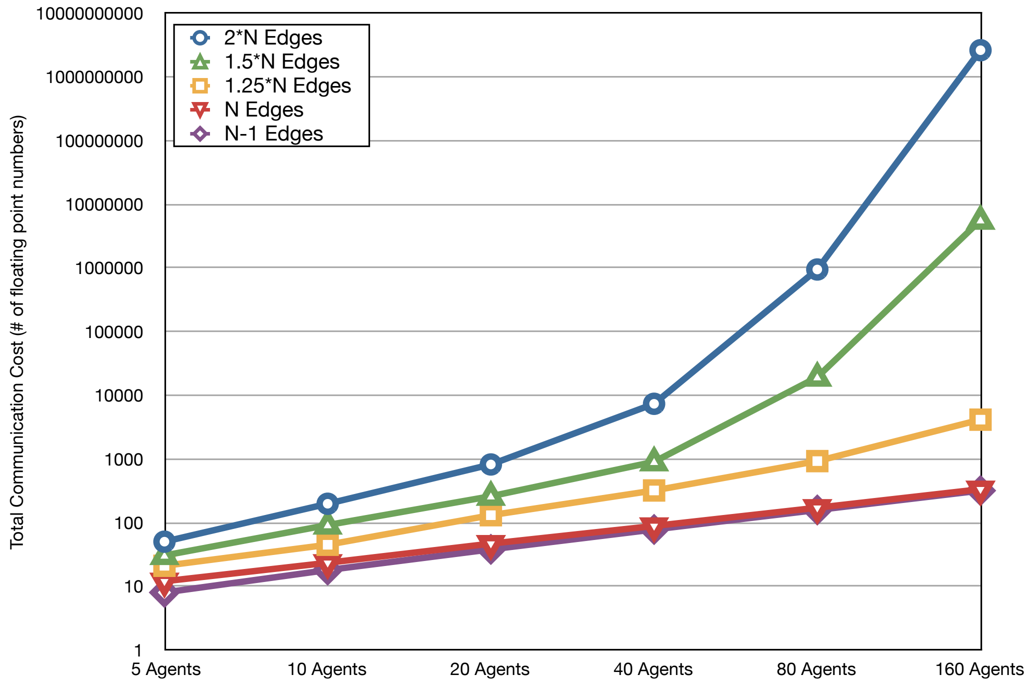


FIGURE 35. Scalability of AEBN as number of agents and edges increases.

When the number of edges is $1.5 * N$ or greater, the cost grows rapidly for networks greater than 40 agents. If we assume floating point numbers are represented with 64 bits, the highest average cost is $\frac{2609853163*64}{1024} = 163,115,823KB$ when a network has 160 agents and 320 edges. This communication cost is highly impractical with current networking technology. The next highest communication cost is $349,767KB$ when the network has 160 agents and 240 edges, which is more practical, but still too high for a large class of networks. When there are 160 agents with 200 edges, the cost is only $258KB$ which is highly practical. These experimental results indicate that when the communication graph is sparse ($\leqslant 1.25 * N$ edges) an AEBN system scales fairly well.

4.10. Extended Simulation

To illustrate some of the flexibility inherent in AEBNs that is not present in MSBNs, consider an extended simulation where we wish to incorporate information from the Center for Disease Control (CDC). The CDC tracks outbreaks of diseases and is responsible for assessing and controlling risks to US citizens. Figure 36 depicts an extended agent communication graph where a CDC threat agent communicates risk assessment of disease outbreaks to the human early indicator agents. The early indicator agents incorporate this information to adjust their sensitivity of monitoring for early indications of disease in human populations. The resulting extended communication graph is no longer a tree and has multiple node disjoint paths between the incident agent and CDC threat agent. The node disjoint paths introduce rumors into the AEBN system which can be removed using the communication solution defined in Section 3.3.

The extended communication graph is not possible for an MSBN system, since the agent graph must be a tree. In order to incorporate the CDC threat agent, the agent communication graph would need to be reorganized into a less natural or undesirable communication graph. Physical restrictions of the distributed nature of the system may make this restructuring costly, inefficient or even impossible. It may be necessary to completely rebuild the MSBN to ensure soundness of sectioning is maintained and that a valid link tree is created for agent communication. Without a global Bayesian network model as a starting point for creating the MSBN system, the difficulty of incorporating new agents would increase dramatically.

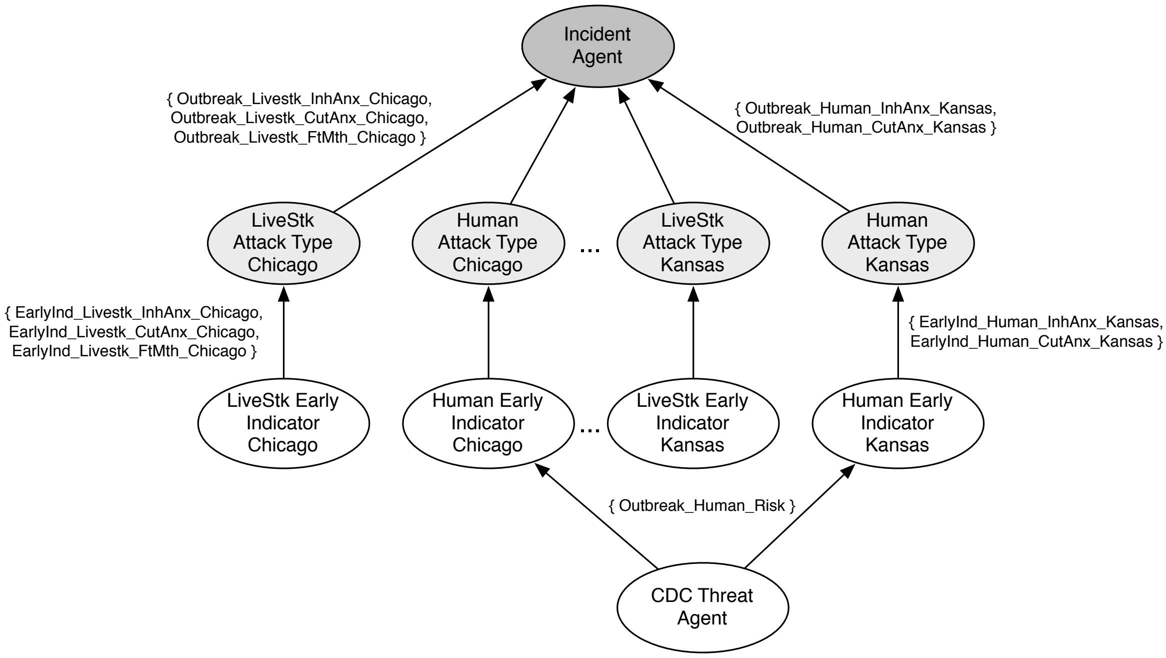


Figure 36. Communication graph for expanded bio-attack AEBN simulation.

5. CONCLUSION

A central goal of this research was to allow easier design of probability-based agents and multiagent systems, resulting in rational decision making. A multiagent framework was presented and compared with other proposed frameworks where advantages and disadvantages of each are outlined. A central problem of message passing in probabilistic systems is the familiar rumor problem, where cycles in message passing cause redundant influence of beliefs. We developed an algorithm to identify the rumor problem in the context of our multiagent system. We characterized the problem of coherence and proposed a design restriction and runtime solution to ensure global coherence. Under the assumption of coherence, we proved the correctness of a runtime algorithm for compensation of rumors. Central to our message passing scheme is the notion of soft evidential update. To evaluate our multiagent model, we devised a simulation that we implemented as an AEBN and MSBN system to compare quantitatively the two formalisms. Finally, we analyzed the scalability of our agent model.

We have identified three avenues for further research. First, we intend to investigate AEBN communication optimizations. To lower communication costs in the multiagent system, the communication graph can be analyzed and redundant communication links could be removed. This situation can occur in the redundancy graph, where expanded messages render some message passing unnecessary. The passing of large joint probability tables between agents in the multiagent system is very expensive, and it may be possible to decompose the messages into a factorized representation that requires far less communication overhead during message passing.

Second, implementation issues of AEBNs should be explored such as dynamic multiagent networks, handling of communication failures, and resolving inconsistent or conflicting evidence.

Third, we want to characterize the joint probability distribution of shared variables represented by an AEBN system. We proved each agent can remove redundant information from received messages using the communication solution. However, proving these beliefs are consistent with a joint probability distribution that is compactly represented by the combined AEBNs is challenging due to the asymmetric nature of the Oracular assumption. We intend to bring to bear recent research in two related areas: algebraic representation of identifiability in causal Bayesian networks, such as (Garcia-Puente et al., 2010), which shows promise in representing a joint probability distribution with asymmetric constraints; conditional random fields (Lafferty et al., 2001), which may also provide a concise way to represent asymmetric constraints. These recent results should be explored further to prove stronger properties of AEBNs and solving the rumor problem.

REFERENCES

Mark Bloemeke. *Agent Encapsulated Bayesian Networks.* PhD thesis, Department of Computer Science, University of South Carolina, 1998.

Hei Chan and Adnan Darwiche. On the revision of probabilistic beliefs using uncertain evidence. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 99–105, Acapulco, Mexico, 2003.

Hei Chan and Adnan Darwiche. A distance measure for bounding probabilistic belief change. *International Journal of Approximate Reasoning*, 38:149–174, 2005.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms.* The MIT Press, Cambridge, MA, 2003.

Adnan Darwiche. *Modeling and Reasoning with Bayesian Networks.* Cambridge University Press, 2009.

W.E. Deming and F.F. Stephan. On a least square adjustment of a sampled frequency table when the expected marginal totals are known. *Annals of Mathematical Statistics*, 11:427, 1940.

Luis David Garcia-Puente, Sarah Spielvogel, and Seth Sullivant. Identifying causal effects with computer algebra. In *Proceedings of the Twenty Sixth Annual Conference on Uncertainty in Artificial Intelligence (UAI-2010)*, Catalina Island, CA, 2010.

Finn V. Jensen. *An Introduction to Bayesian Networks.* Springer-Verlag, New York, NY, 1995.

Finn V Jensen. *Bayesian Networks and Decision Graphs.* Springer, New York, NY, 2001.

Young-Gyun Kim, Marco Valtorta, and Jiří Vomlel. A prototypical system for soft evidential update. *Applied Intelligence*, 21(1), July–August 2004.

Dexter C. Kozen. *The Design and Analysis of Algorithms.* Springer-Verlag, New York, 1992.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1-55860-778-1. URL `http://portal.acm.org/citation.cfm?id=645530.655813`.

Scott Langevin and Marco Valtorta. Performance evaluation of algorithms for soft evidential update in Bayesian networks: First results. In *Proceedings of the Second International Conference on Scalable Uncertainty Management (SUM-08)*, pages 284–297, Naples, Italy, October 2008.

Scott Langevin, Marco Valtorta, and Mark Bloemeke. Agent-encapsulated bayesian networks and the rumor problem. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1 - Volume 1*, AAMAS '10, pages 1553–1554, Toronto, Canada, 2010. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-0-9826571-1-9. URL `http://portal.acm.org/citation.cfm?id=1838206.1838477`.

Kathryn Blackmond Laskey and Tod Levitt. Multisource fusion for probabilistic detection and opportunistic assessment of terrorist threats. In *Proceedings of Aerosense 2002*, 2002.

L. Motyckova. Maximum flow problem in distributed environment. In *SOFSEM '95: Theory and Practice of Informatics*, pages 425–430, Berlin, 1995. Springer-Verlag.

Rong Pan. *Semantically-Linked Bayesian Network: A Framework for Probabilistic Inference over Multiple Bayesian Networks.* PhD thesis, Department of Computer Science and Electrical Engineering, University of Maryland, 2006.

Rong Pan, Yun Peng, and Zhongli Ding. Belief update in Bayesian networks using uncertain evidence. In *ICTAI*, pages 441–444. IEEE Computer Society, 2006.

Judea Pearl. *Causality: Modeling, Reasoning, and Inference.* Cambridge University Press, Cambridge, 2000.

Simukai W. Utete. Local information processing for decision making in decentralised sensing networks. In *Proceedings of the Eleventh International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE-98)*, pages 667–676, Benicassim, Castellon, Spain, 1998.

Marco Valtorta, Young-Gyun Kim, and Jiří Vomlel. Soft evidential update for probabilistic multiagent systems. *International Journal of Approximate Reasoning*, 29(1):71–106, January 2002.

Y. Xiang. *Multiply sectioned Bayesian belief networks for large knowledge-based systems: an application to neuromuscular diagnosis.* PhD thesis, Electrical and Computer Engineering, University of British Columbia, 1992.

Y. Xiang and V. Lesser. Justifying multiply sectioned Bayesian networks. In *Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS-2000)*, pages 349–356, Boston, MA, July 2000.

Y. Xiang and V. Lesser. On the role of multiply sectioned Bayesian networks to cooperative multiagent systems. *IEEE Trans. Systems, Man, and Cybernetics-Part A*, 33:489–501, 2003.

Yang Xiang. A probabilistic framework for cooperative multi-agent distributed interpretation and optimization of communication. *Artificial Intelligence*, 86:295–342, 1996.

Yang Xiang. *Probabilistic Reasoning in Multiagent Systems: A Graphical Models Approach.* Cambridge University Press, Cambridge, 2002.

Yang Xiang and Franklin Hanshar. Comparison of tightly and loosely coupled decision paradigms in multiagent expedition. *Int. J. Approx. Reasoning*, 51(5):600–613, 2010. ISSN 0888-613X. .

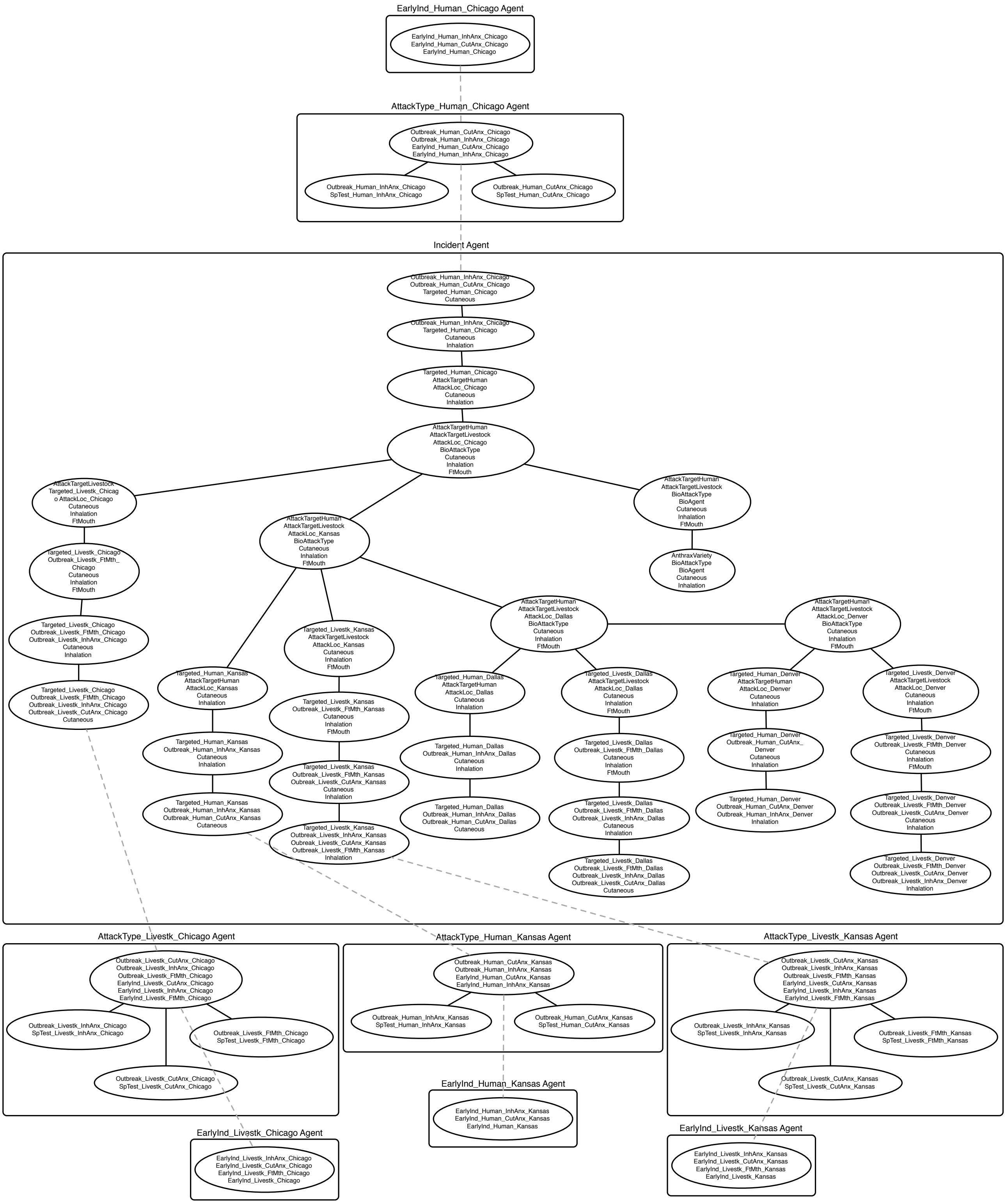


FIGURE 37. Linked Junction Forest for bio-attack MSBN simulation (only Chicago and Kansas agents shown).