Agent Encapsulated Bayesian Networks—DRAFT☆

Scott Langevin[a], Marco Valtorta[b,*]

[a]*Oculus Info Inc., 2 Berkeley Street, Suite 600, Toronto, Ontario, Canada M5A 4J5*
[b]*University of South Carolina, 315 Main Street, Columbia, SC, USA 29208*

Abstract

In this research, we define a cooperative multiagent system where the agents use locally designed Bayesian networks to represent their knowledge. 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. We compare and contrast our system with the MSBN multiagent model.

Finally, we implement a multiagent system for experimentation using our multiagent system and MSBNs. We devise performance measures to compare the two systems. From this comparison, we provide guidance for the design of probabilistic multiagent systems.

Keywords: Multiagent Systems, Bayesian networks, Distributed Artificial Intelligence

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

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

*Corresponding author

Email addresses: scott.langevin@oculusinfo.com (Scott Langevin), mgv@cec.sc.edu (Marco Valtorta)

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

In this paper, we provide clear assumptions about agents that use probabilistic representations of knowledge, guidelines for their design, and efficient algorithms for communicating (or sharing) probabilities. The goal is to allow easier design of probability-based agents and multiagent systems, resulting in rational decision making. 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.

The rest of this paper is organized as follows...

2. Agents and Multiagent systems

An *agent* is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators [1, 2]. The agent's behavior is described by an agent function that takes as input the percepts from the agent's sensors and outputs the actions the agent takes in the environment via its actuators (1).

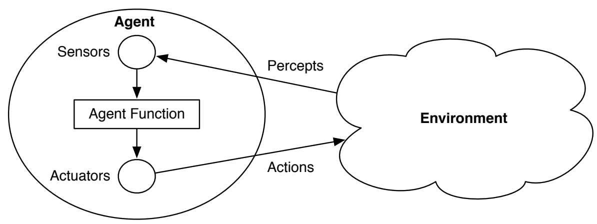

Figure 1: Conceptual agent model as presented in [1].

This abstraction is useful since it describes living organisms such as humans, as well as computational agents that encompass virtual (software) and robotic (hardware) agents.

Intelligent agents are agents that act rationally by choosing actions that are expected to achieve its goals or maximize its utility. The specific features that define an intelligent agent are: (1) it has an internal model that represents its prior knowledge and state of the world it is designed for; (2) it has a set of goals or preferences of the state of the world; (3) it senses the environment it operates in and updates its internal model to be consistent with the observations; and (4) it takes actions based on its belief in the state of the world and that are expected to achieve its goals. We say the agent reasons about the state of the world to produce desirable outcomes.

In this paper we only consider intelligent computational agents, and not natural agents such as people, animals, etc. From here onwards we will simply refer to intelligent computational agents as agents.

It follows from the description of an intelligent agent that the three main tasks of an agent are:

1. *Sensing*: Using its sensors to make observations
2. *Reasoning*: Reasoning about the true state of the environment
3. *Acting*: Take actions based on goals and belief of true state of environment

Our research focuses on the first two tasks, where an agent reasons about the state of the world based on evidence it receives either from local sensors or messages from other agents. Whether the agents utilize their internal models using goal based planning or the principle of maximum expected utility is outside the scope of this paper.

A *multiagent system* (MAS) is characterized as a system composed of more than one agent where the agents communicate directly or indirectly through interactions in the environment.

Often the domain world of the agent is too large for one single agent to sense and act upon the state of the world to achieve its goals. For example, consider an intrusion detection system for a large corporate computer network system. A single agent monitoring the network would need to monitor each servers logs, firewalls, routers, incoming and outgoing network connections, email systems, database accesses, web server traffic, etc. The agent would need to have details on various particular vendors of these systems so as to be general enough to be a commercial product. This is a very challenging task and centralizing these tasks would require high computational and communication cost as well as agent complexity. To manage the complexity, the tasks could be divided up amongst several agents rather than one large agent. The process is to modularize a system into separate functional units and assign a functional unit to an agent. The agents communicate with each other to share information in order to achieve the overall system goal.

In a multiagent system, agent communication is often necessary since observations that are useful may be sensed by an agent that has direct access to the required sensor, and other agents may require this information. The observing agent can communicate the observation to the interested agents. Observations that are directly sensed by an agent are called local observations (evidence) and observations that are from other agents are called external observation. The observations are often referred to as evidence.

When designing a multiagent system the following key considerations must be addressed [3] :

- How do agents represent knowledge?
- How do agents communicate?
- Whom should agents be allowed to communicate with?
- What is the purpose of the communication?

- How should messages be processed?
- Is global consistency desired, and if so how is global consistency maintained?

These questions will be addressed in this paper with the proposal of a multiagent system where the agents represent their knowledge using Bayesian networks and communicate via messages that represent shared beliefs. The details of this agent model will be discussed further in the following chapters.

Agents can be either cooperative or selfish. Cooperative agents do not attempt to deceive or gain advantage over other agents, rather they work with other agents to achieve mutual goals. Selfish agents, however, will act in their best interest at the expense of other agents. In this paper, we only consider cooperative agents that work together to reason over the state of the world.

3. Related Work

In this section, we briefly review related work in the area of distributed Bayesian networks and multiagent systems. In the following sections we will review probabilistic multiagent systems and present our agent model.

3.1. Distributed Bayesian Networks

Multi-Entity Bayesian Network (MEBN) [4, 5], proposed by Laskey et al. extends the expressiveness of Bayesian networks through the instantiation of parameterized Bayesian network fragments (MFrags) that are composed into a situation-specific Bayesian network (SSBN), using a knowledge-based network construction mechanism. The MEBN framework allows for a distributed representation of a Bayesian network, but after the SSBN is constructed, reasoning is centralized using standard Bayesian inference. Therefore, MEBN is not appropriate for multiagent distributed interpretation tasks.

Semantically-Linked Bayesian Networks (SLBN) [6] is a framework for performing probabilistic inference over individually designed Bayesian networks. SLBN defines linkages between semantically similar variables where probabilistic influences are propagated over variable linkages from one Bayesian network to another using soft evidence and virtual evidence. Soft evidential update is performed using the wrapper methods of [7].

SLBN is proposed as a general distributed inferencing method, and does not address use in a multiagent environment. Further, SLBN does not address the problem when linkages result in multiply connected Bayesian networks. Therefore, the rumor problem [8, 9] is not solved or addressed.

The advantage of SLBN over our proposed work is the lack of need for a common ontology. However, SLBN requires a mapping function to related similar concepts amongst the linked Bayesian networks. Our work assumes this mapping has been implicitly performed during the design of each AEBNs internal model.

In the next section, we discuss distributed Bayesian network models under a multiagent paradigm, and propose our model.

3.2. Truth Maintenance Systems

A justification-based truth maintenance system (JTMS) is a form of non-monotonic belief revision system where the beliefs are restricted to propositional logic representations. In a JTMS, inferred assertions are dependent on the direct assertions used to assert it during inferencing. Each inferred assertion maintains a "justification set" of assertions that justify the inferred assertions truth. When new facts are introduced or retracted, logical consistency is maintained by introducing or retracting assertions. Each assertion that has a relationship to the new facts, update their justification set by adding or removing assertions to be consistent with the new facts. If the justification set becomes empty then the proposition is labeled OUT to represent that it no longer holds in the JTMS knowledge base. If they do hold, they are labeled IN.

JTMS is similar to hard and soft evidential update in Bayesian networks, in that the belief system is revised to respect the new evidence and be consistent with it. This is a non-monotonic belief update, where previously held beliefs may be reduced or increased based on the new evidence. In the case of JTMS, the beliefs must be 0 or 1, while a Bayesian network is more general and allows degrees of beliefs in [0,1]. Since each assertion is dependent on a set of justifying assertions in the JTMS, we can represent the dependence using a DAG, where parent nodes represent direct causal relationships to the children nodes. This is analogous to the dependence structure of causal Bayesian networks and means we could represent a JTMS as a Bayesian network where the conditional probability tables represent the disjunction of the supporting propositions in the JTMS. Each node represents a proposition and has two states: IN and OUT to represent whether the proposition holds in the knowledge base. To perform belief update, new facts can be entered as hard evidence in the Bayesian network and standard belief propagation can be performed to revise the knowledge base.

Belief update in Bayesian networks with virtual evidence is a different method of incorporating uncertain evidence. It does not hold the same properties as hard and soft evidence where the evidence is treated as a constraint on final beliefs. Virtual evidence only specifies the strength a piece of evidence has on a belief system and therefore is not treated as a constraint on the final beliefs. Only the likelihood ratio of the virtual evidence is respected. In this way belief update with virtual evidence does not correspond to JTMS.

Multiagent Truth Maintenance (MTM) proposed by Huhns and Brigeland [10] is a cooperative multiagent framework where each agent reasons non-monotonically using justification-based truth maintenance systems (JTMS) to maintain the integrity of its own knowledge base. Agents communicate by exchanging messages that are assertions over shared variables. A distributed truth maintenance algorithm determines whether global inconsistencies amongst agents need to be resolved and provides a mechanism for resolving them. The cost of ensuring complete global consistency of all the agents' knowledge bases makes this property impractical. Instead MTM aims for a weaker form of global consistency by ensuring consistency of shared data amongst agents.

Our proposed mutiagent system shares similar belief revision goals as MTM, such as maintaining global consistency of shared beliefs amongst the agents. In both systems, agents partition their internal models into private and shared variables. The shared beliefs are exchanged between agents via message passing, while the private variables are internal to an agent. Agents receiving messages revise their internal models to be consistent with the received beliefs. The belief revision process in both models commit to a principe of minimizing the change of belief necessary to ensure consistency.

MTM differs from AEBNs in that a receiving agent can reject received information if it conflicts with strongly held local beliefs. In AEBN, we assume publishing agents have oracular or expert knowledge over their shared variables and subscribing agents must respect the publishers beliefs. However, it is permitted for an AEBN to discount a publishing agent's belief by explicitly modeling the agent's reliability directly into its internal knowledge model. Additionally, the publisher/subscriber nature of the AEBN system restricts agent communication to a DAG topology, whereas in MTM communication can be bidirectional or contain cycles.

Finally, in an MTM multiagent system reasoning is based on propositional logic and therefore the system does not support reasoning about uncertainty, while our multiagent model is based on uncertain reasoning using Bayesian networks. While there are similarities in the belief revision methods of the two multiagent systems, they are essentially aimed at addressing completely different problem domains.

4. Probabilistic Multiagent Systems

This section is concerned with multiagent systems that represent their knowledge using probability and share their beliefs with other agents in the system. We review a prominent approach to probabilistic knowledge representation in multiagent systems, and then introduce our agent model.

As Pan identifies, the three main issues of probabilistic multiagent systems are [6]:

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. In this paper, we will highlight some of the differences between our agent model and others proposed.

4.1. Multiply Sectioned Bayesian Networks

Multiply Sectioned Bayesian Networks (MSBNs) [3, 11] 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 [3, 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 [3].

Definition 4.1 (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 4.2 (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 [12]. 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 [13, 14, 11, 3, 15], 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 [13], 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. Our agent model allows for more flexible topologies than trees, but when the agent topology is multiply connected, global consistency can only be achieved by detecting and compensating for what we call rumors to avoid bias. The details of solving the rumor problem are outside the scope of this paper and can be found in [8, 9].

5. 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.

5.1. Original Formalism

This research extends the Agent Encapsulated Bayesian Network (AEBN) multiagent model originally proposed by Bloemeke [16]. 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 2(a). Since only single marginals are passed, agent Y will receive $P(A)$ and $P(B)$ losing any dependence between them.

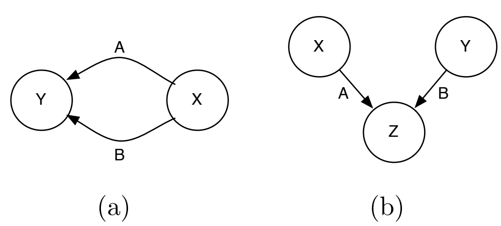

Figure 2: Loss of dependence between variable A and B.

In the second example, agent X and Y send their beliefs in variables A and B respectively to agent Z, shown in Figure 2(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. [17] argue that agents should not force independence of the received evidence, and illustrate their argument with the following example[1]:

Example 5.1. *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,* $P(A, E)$.

[1]The example was inspired by Demming and Stephans 1940 paper on IPFP [18].

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 has recently been the subject of methological inquiry and algorithm development [17, 19, 20, 21, 22, 23, 7, 24] can be used to satisfy these requirements by revising an agent's joint probability distribution to respect the beliefs received from publishing agents.

5.2. 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 5.1 (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 [17], 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 [17, 19]. Therefore, each agent that receives messages from other agents obtains soft evidence for one or more observation variables[2] (see Figure 3) 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. 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

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

$P(Obs_i = o|\pi(Obs_i) = \vec{x}) = 1 \Longleftrightarrow \vec{x}$ corresponds to o.

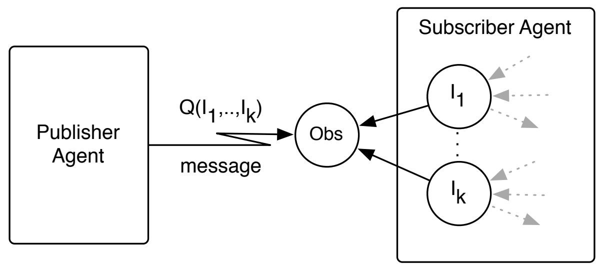

Figure 3: Introduction of an observation variable in a subscriber agent for absorption of a publisher message over shared variables $I_1, ..., I_k$.

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 [17, Section 5] for a detailed discussion on this point.) Lemma 1 in [17] 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 [17, 20], *lazy big clique algorithm* of [25], or the *wrapper methods* of [7].

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 5.3.

5.3. 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 [26]. 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 [26, 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 4 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.

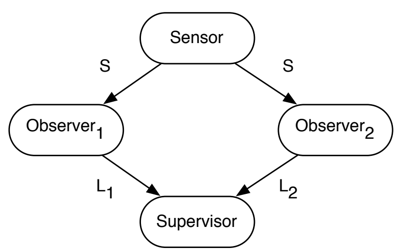

Figure 4: Redundantly Observed Sensor Example (ROSE) communication graph.

6. Multiagent Simulation

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.

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 [27]. 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 [27]:

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

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[3] to illustrate the power of MEBNs, we constructed a similar Bayesian network (Figure 5) 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.

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 6 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

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

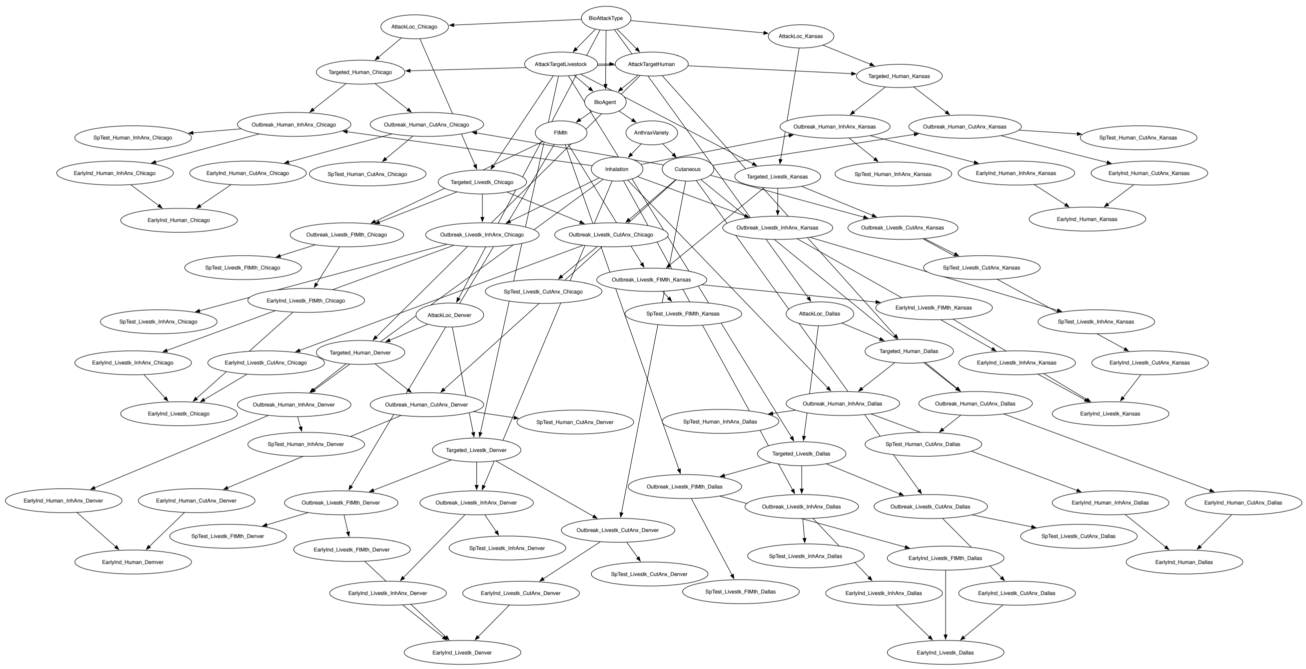

Figure 5: Bio-attack full Bayesian network.

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 6.2.

6.0.1. 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.

6.0.2. 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 in-

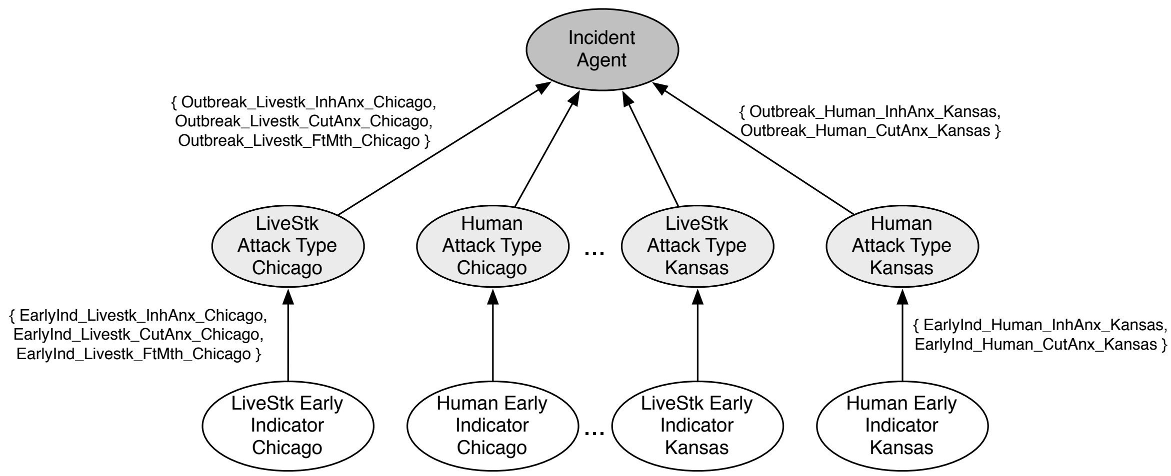

Figure 6: Communication graph for bio-attack AEBN simulation (only Chicago and Kansas agents shown).

halation, 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.

6.0.3. 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.

6.1. 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 7. An example of the Bayesian networks for the attack type agents is shown in Figure 8 and Figure 10 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 9 and Figure 11 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 5) 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 [4] 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 6.3.

6.2. 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[5] to construct and validate the soundness of sectioning (see Section 4.1 for details) of the MSBN. Figure 12 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.

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,

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

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

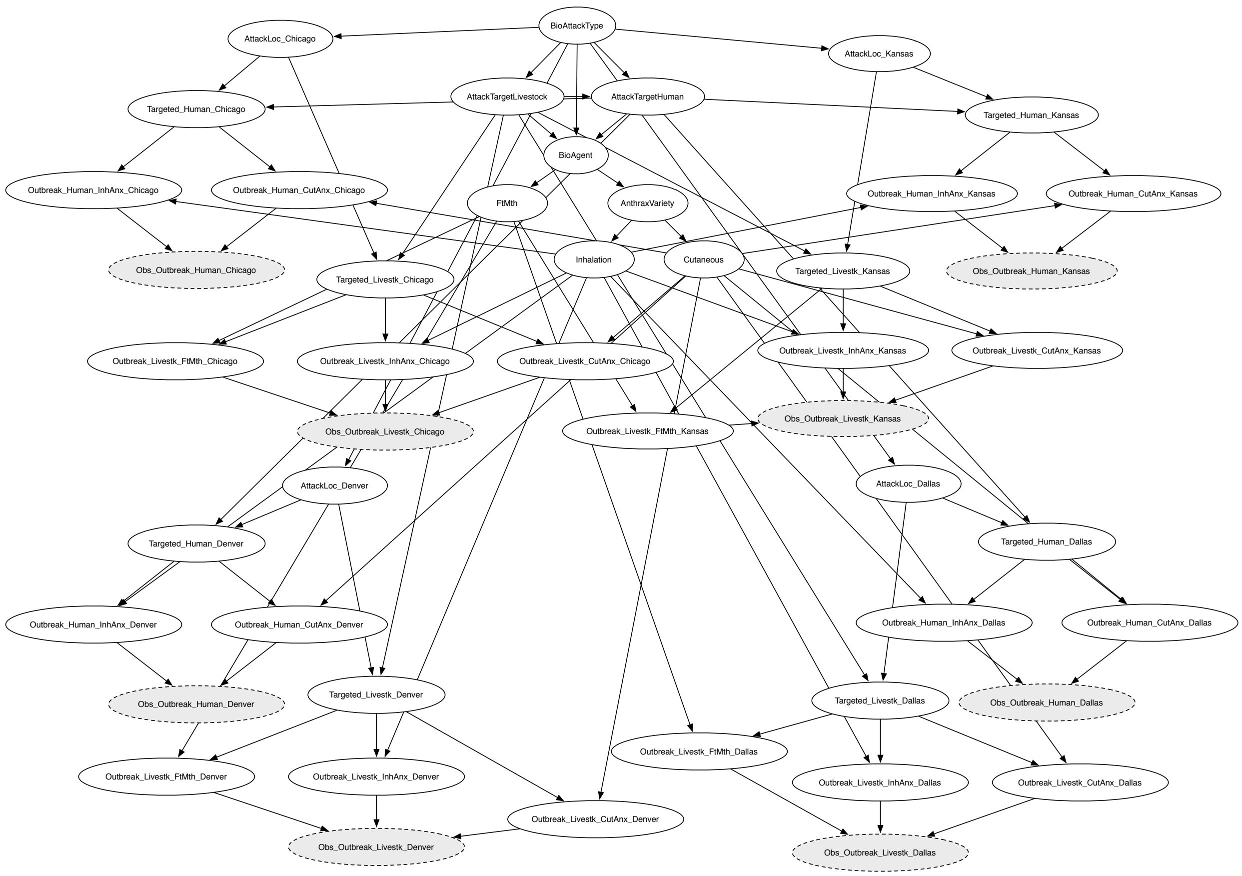

Figure 7: Bayesian network for Incident agent.

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 6.3. As with the AEBN simulation, we assume perfect communication in the MSBN simulation.

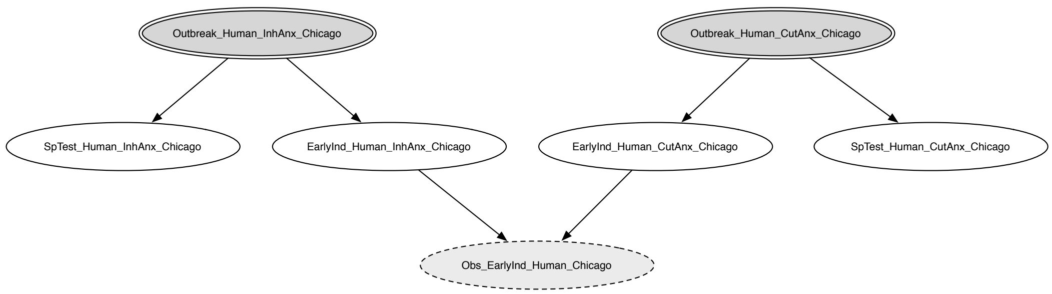

Figure 8: Bayesian network for Chicago Human Attack Type agent.

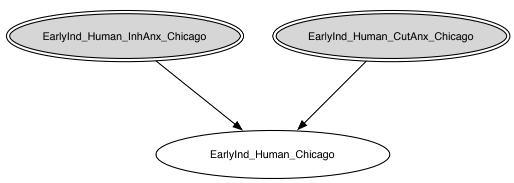

Figure 9: Bayesian network for Chicago Human Early Indicator agent.

6.3. 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 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.

6.4. 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

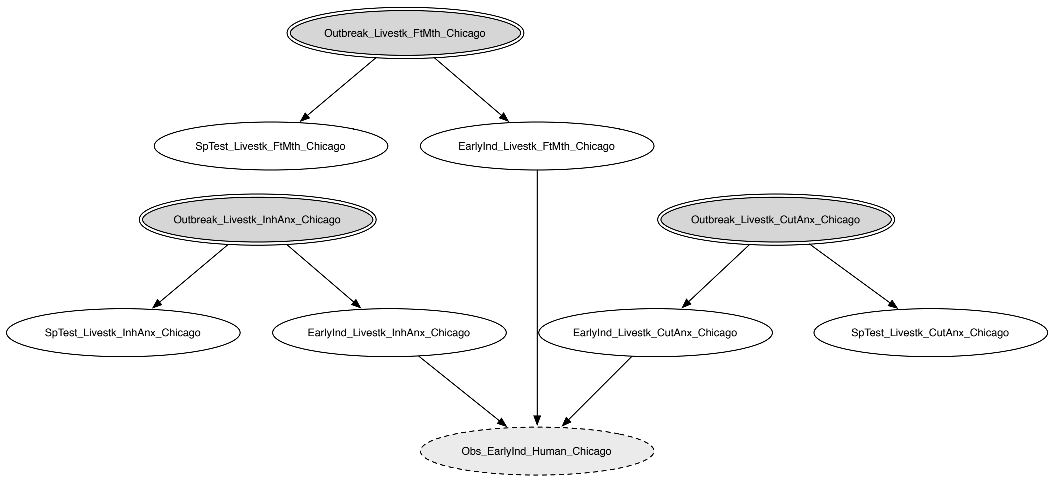

Figure 10: Bayesian network for Chicago Livestock Attack Type agent.

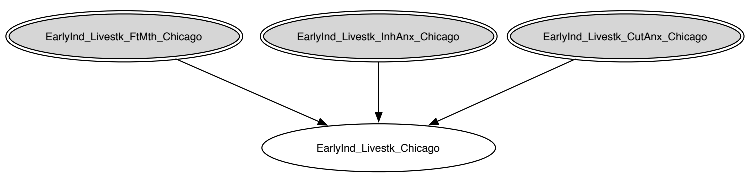

Figure 11: Bayesian network for Chicago Livestock Early Indicator agent.

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 13 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.

In Figures 14 and 15 we show the revised Bayesian network models of the Chicago human 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.

7. 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 16, 17, 18 and 19 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.

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[6] of MSBN and AEBN v2. Only AEBN v2 was compared to MSBN since the plots in Figure 16-19 indicate it is overall superior to AEBN v1. The resulting distance measure results are shown in Figure 20 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.

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

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

belief difference.

Finally, Figure 21 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 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.

8. Conclusions and Future Work

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. To evaluate our multiagent model, we devised a simulation that we implemented as an AEBN and MSBN to compare quantitatively the two formalisms.

8.1. Future Work

During the course of our research we have identified several possible avenues for further research:

1. Investigate AEBN communication optimizations.
   (a) To lower communication costs, 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.
   (b) The passing of large joint probability tables between agents 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.

2. Implementation issues of AEBNs should be explored such as dynamic multiagent networks, handling of communication failures, and resolving inconsistent or conflicting evidence.
3. Characterize the joint probability distribution of shared variables represented by an AEBN system.
   (a) In our research, 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. Recent research in identifiability in causal Bayesian networks, such as [28, 29] shows promise in representing a joint probability distribution with asymmetric constraints. These recent results should be explored further to prove stronger properties of AEBNs and solving the rumor problem.

References

[1] S. J. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, 2nd Edition, Pearson Education, New Jersey, 2003.

[2] D. Poole, A. Mackworth, Artificial Intelligence: Foundations of Computational Agents, Cambridge University Press, Cambridge, 2010.

[3] Y. Xiang, Probabilistic Reasoning in Multiagent Systems: A Graphical Models Approach, Cambridge University Press, Cambridge, 2002.

[4] K. B. Laskey, S. M. Mahoney, E. Wright, Hypothesis management in situation-specific network construction, in: Proceedings of the Seventeenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-01), Seattle, WA, 2001, pp. 301–309.

[5] K. B. Laskey, MEBN: A language for first-order Bayesian knowledge bases, Artificial Intelligence 172 (2-3) (2008) 140 – 178. doi:DOI: 10.1016/j.artint.2007.09.006.
URL `http://www.sciencedirect.com/science/article/B6TYF-4PTMXXP-1/2/ce6bcf1c5a`

[6] R. Pan, Semantically-linked Bayesian network: A framework for probabilistic inference over multiple Bayesian networks, Ph.D. thesis, De-

partment of Computer Science and Electrical Engineering, University of Maryland (2006).

[7] R. Pan, Y. Peng, Z. Ding, Belief update in Bayesian networks using uncertain evidence, in: ICTAI, IEEE Computer Society, 2006, pp. 441–444.

[8] S. Langevin, M. Valtorta, M. 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, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2010, pp. 1553–1554.

[9] S. Langevin, Knowledge representation, communication, and update in probability-based multiagent systems, Ph.D. thesis, Department of Computer Science and Engineering, University of South Carolina (2011).

[10] M. Huhns, D. Bridgeland, Multiagent truth maintenance, Systems, Man and Cybernetics, IEEE Transactions on 21 (6) (1991) 1437 –1445. doi:10.1109/21.135687.

[11] Y. Xiang, V. Lesser, Justifying multiply sectioned Bayesian networks, in: Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS-2000), Boston, MA, 2000, pp. 349–356.

[12] Y. Xiang, Multiply sectioned Bayesian belief networks for large knowledge-based systems: an application to neuromuscular diagnosis, Ph.D. thesis, Electrical and Computer Engineering, University of British Columbia (1992).

[13] Y. Xiang, F. Hanshar, Comparison of tightly and loosely coupled decision paradigms in multiagent expedition, Int. J. Approx. Reasoning 51 (5) (2010) 600–613. doi:http://dx.doi.org/10.1016/j.ijar.2010.01.016.

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

[15] Y. Xiang, A probabilistic framework for cooperative multi-agent distributed interpretation and optimization of communication, Artificial Intelligence 86 (1996) 295–342.

[16] M. Bloemeke, Agent encapsulated Bayesian networks, Ph.D. thesis, Department of Computer Science, University of South Carolina (1998).

[17] M. Valtorta, Y.-G. Kim, J. Vomlel, Soft evidential update for probabilistic multiagent systems, International Journal of Approximate Reasoning 29 (1) (2002) 71–106.

[18] W. Deming, F. Stephan, On a least square adjustment of a sampled frequency table when the expected marginal totals are known, Annals of Mathematical Statistics 11 (1940) 427.

[19] H. Chan, A. Darwiche, On the revision of probabilistic beliefs using uncertain evidence, in: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, 2003, pp. 99–105.

[20] Y.-G. Kim, M. Valtorta, J. Vomlel, A prototypical system for soft evidential update, Applied Intelligence 21 (1).

[21] J. Vomlel, Probabilistic reasoning with uncertain evidence, Neural Network World, International Journal on Neural and Mass-Parallel Computing and Information Systems 14 (5) (2004) 453–456.

[22] H. Chan, A. Darwiche, On the revision of probabilistic beliefs using uncertain evidence, Artificial Intelligence 163 (2005) 67–90.

[23] Y. Peng, Z. Ding, Modifying Bayesian networks by probability constraints, in: Proceedings of the Twenty-first Annual Conference on Uncertainty in Artificial Intelligence (UAI-05), Edinburgh, Scotland, 2005, pp. 459–466.

[24] E. Di Tomaso, J. Baldwin, An approach to hybrid probabilisitic models, International Journal of Approximate Reasoning 47 (2008) 202–218.

[25] S. Langevin, M. 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), Naples, Italy, 2008, pp. 284–297.

[26] J. Pearl, Causality: Modeling, Reasoning, and Inference, Cambridge University Press, Cambridge, 2000.

[27] K. B. Laskey, T. Levitt, Multisource fusion for probabilistic detection and opportunistic assessment of terrorist threats, in: Proceedings of Aerosense 2002, 2002.

[28] Y. Huang, M. Valtorta, Identifiability on causal Bayesian networks: A sound and complete algorithm, in: Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06), 2006, pp. 1149–1154.

[29] L. D. Garcia-Puente, S. Spielvogel, S. 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.

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.**

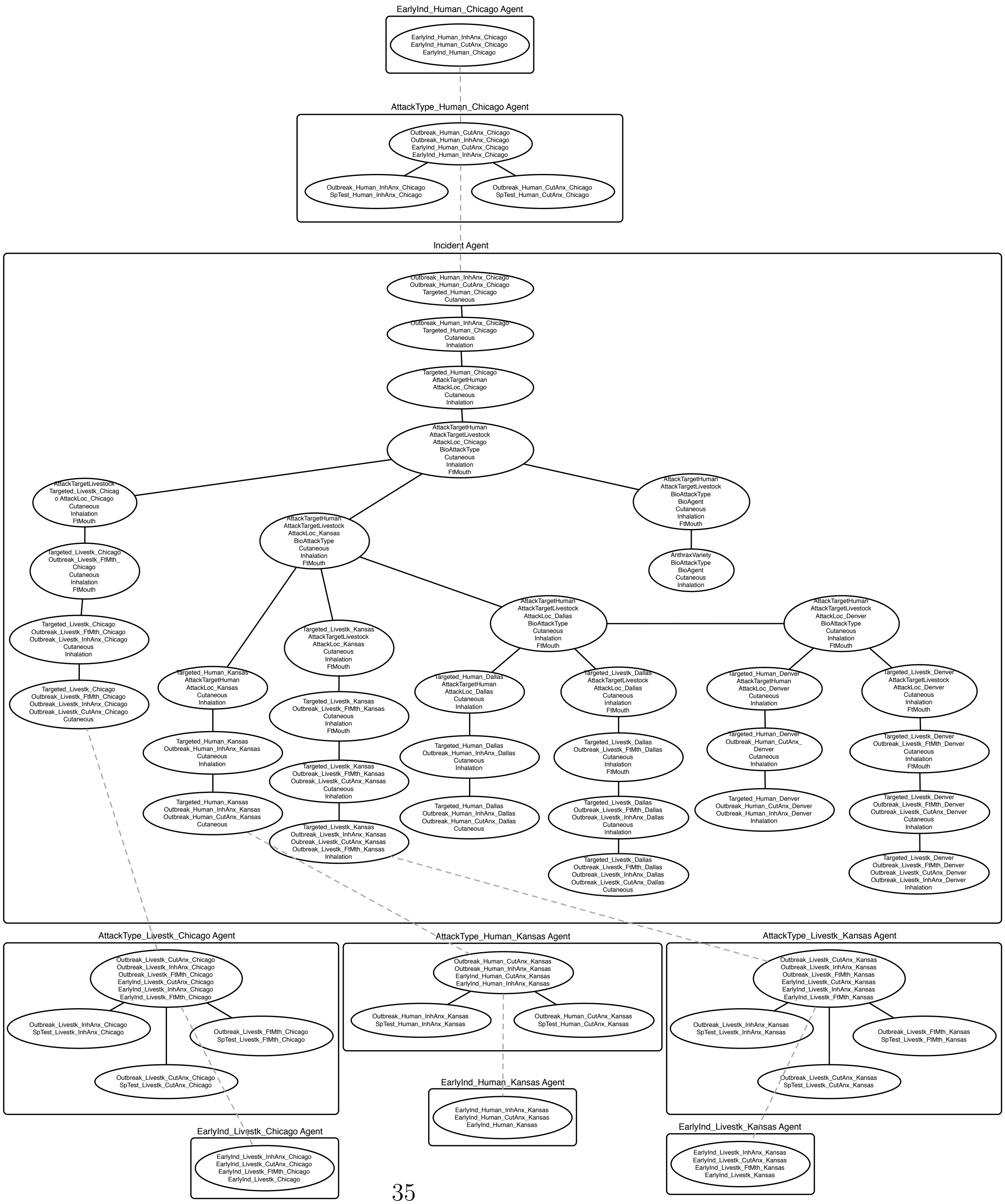

Figure 12: Linked Junction Forest for bio-attack MSBN simulation (only Chicago and Kansas agents shown).

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

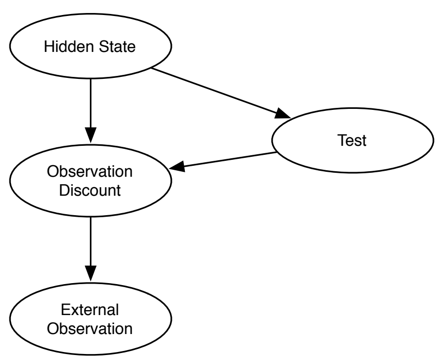

Figure 13: Discounting external evidence given an accurate test modeling technique.

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

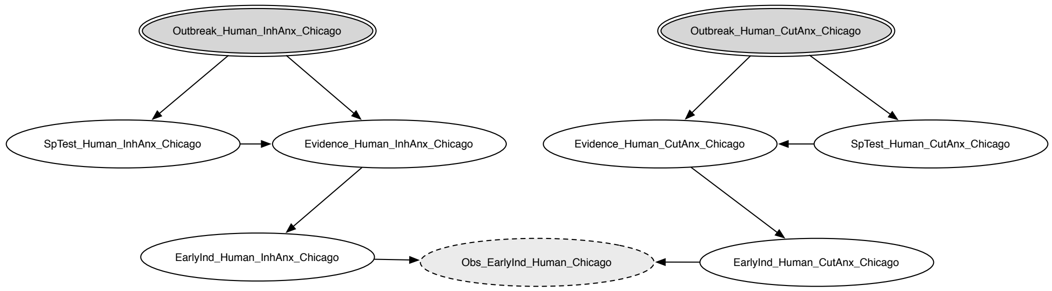

Figure 14: Revised Bayesian network for Chicago Human Attack type agent.

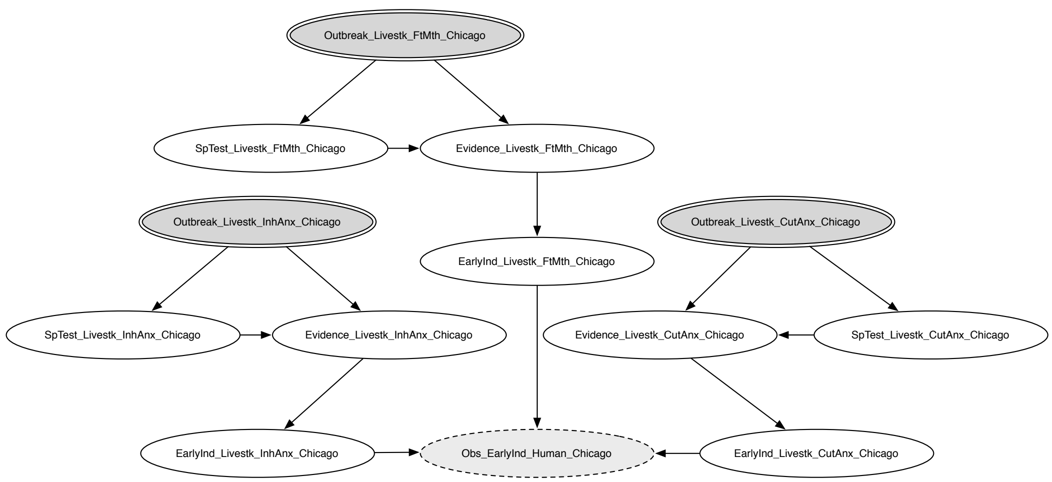

Figure 15: Revised Bayesian network for Chicago Livestock Attack type agent.

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

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

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

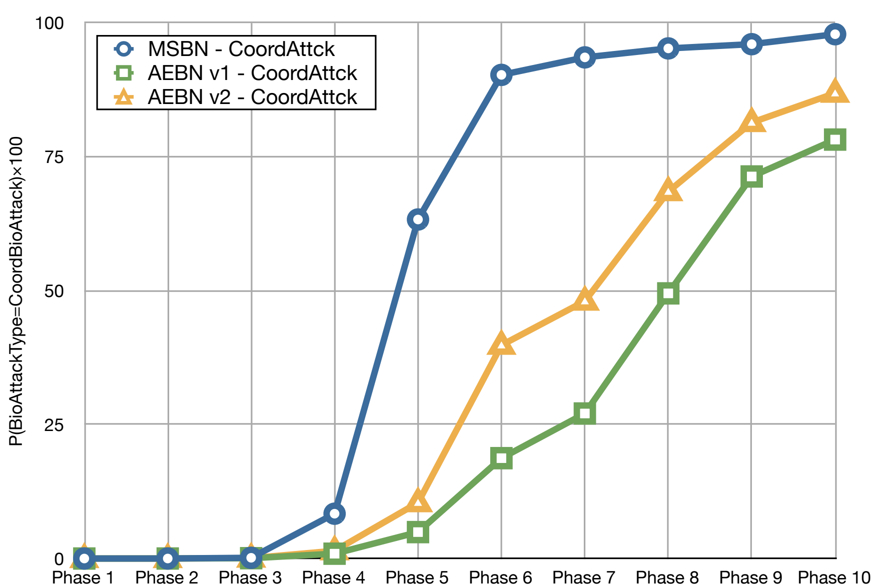

Figure 16: Beliefs of Coordinated Bio Attack over scenario phases.

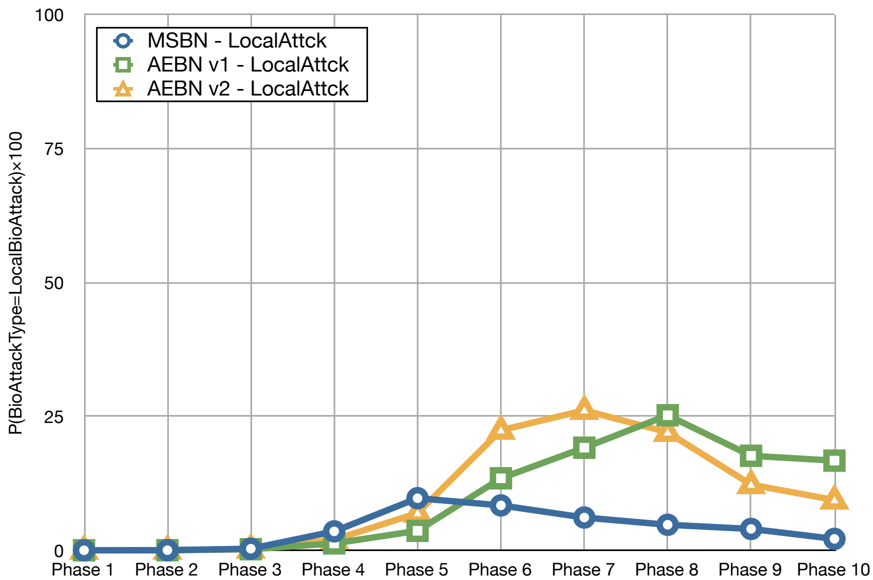

Figure 17: Beliefs of Local Bio Attack over scenario phases.

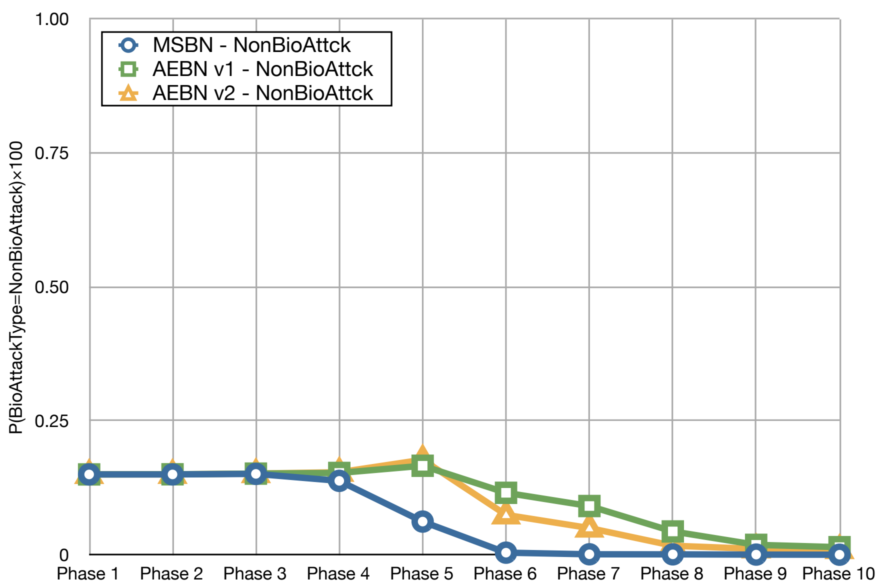

Figure 18: Beliefs of Non-Bio Attack over scenario phases.

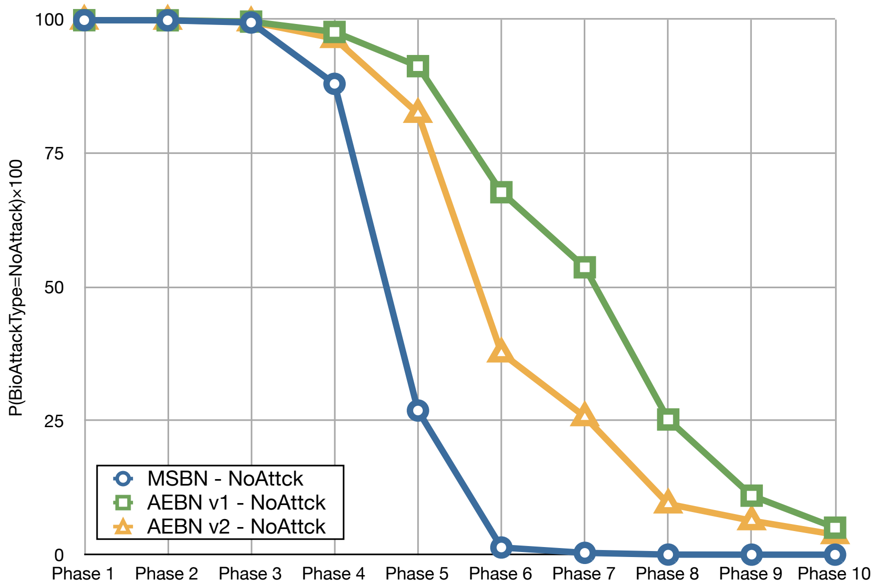

Figure 19: Beliefs of No Attack over scenario phases.

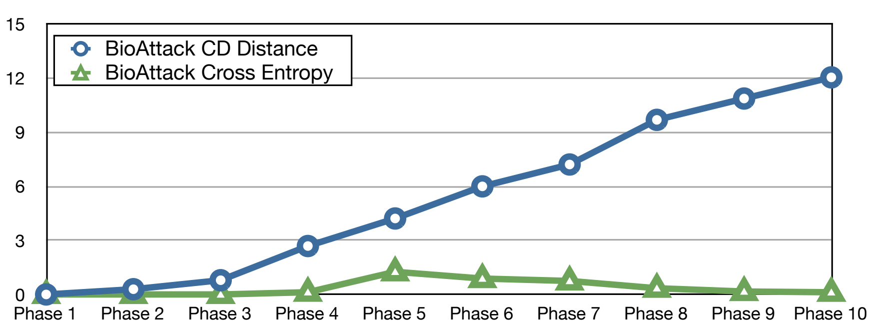

Figure 20: CD and I-divergence of BioAttackType distribution in AEBN v2 and MSBN over scenario phases.

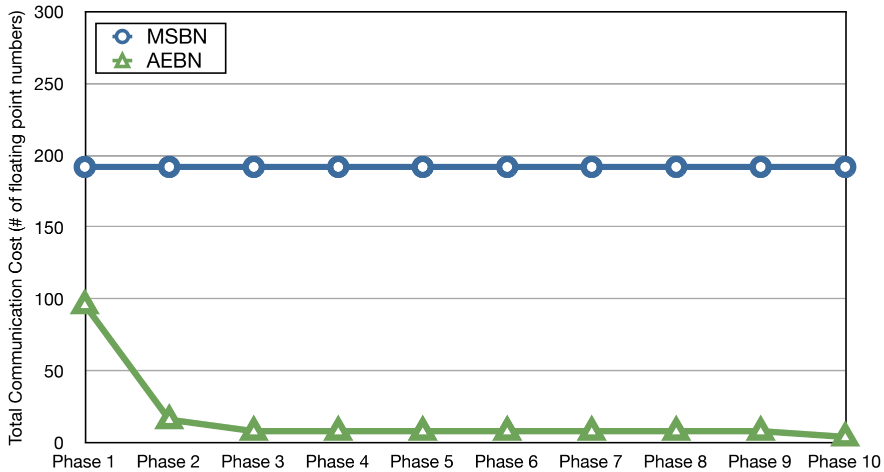

Figure 21: Total communication cost of AEBN v2 and MSBN over scenario phases.