Constructing Bayesian Network in a Changing World

Yanping Xiang **Kim-Leng Poh**

Department of Industrial & Systems Engineering
National University of Singapore
10 Kent Ridge Crescent
Singapore 119260
{iseyp, isepohkl}@nus.edu.sg

Abstract

Although the process of decision-making has been investigated for centuries, only in the last few decades have investigators systematically addressed how decisions are made in a dynamic, real-world environment. One of the most daunting challenges faced by decision support systems is a perpetual change in their environment. Existing decision support methodologies, tools, and frameworks are often difficult to scale up and adapt to changing knowledge. In this paper, a framework that constructs a decision model from knowledge base according to the changes in environment is presented. The generated Bayesian network can be changed as the situation evolves. This paper describes the knowledge representation and model construction process. Bayesian networks combination method is also discussed in details.

Introduction

A crucial problem that decision makers face is the problem of uncertainty and the dynamics of the environment. Decision theory provides a normative framework for representing and reasoning about decision problems under uncertainty. Within the context of this framework, researchers in uncertainty in the AI community have been developing computational techniques for building rational agents and representations suited for engineering their knowledge bases. There exist some methods, often mentioned in artificial intelligence and expert systems literature, which also handle uncertainty quite adequately, via fuzzy logic, belief functions etc. The flexibility of Bayesian networks (BN) for representing probabilistic dependencies and the relative efficiency of computational techniques for performing inference over them makes Bayesian networks an extremely powerful tool for solving problems involving uncertainty. In this paper, Bayesian network models are employed as decision models. In most current applications of Bayesian networks, the domain knowledge is represented by a single network model that applies to all problem instances. In more complex domains however, problem-specific models must be constructed from a knowledge base encoding probabilistic relationships in the domain. In recent years, many knowledge representation schemes have been developed for the construction of probabilistic and decision models (Ngo and Haddawy 1997; Haddawy 1994; Poole 1993; Laskey and Mahoney 1997). This knowledge-based model construction approach has been applied in many domains.

Although the process of decision-making has been investigated for centuries, only in the last few decades have investigators systematically addressed how decisions are made in a dynamic, real-world environment. Decision-making in a static environment is very different from that in a dynamic, real-world environment. One of the most daunting challenges faced by decision support systems is a perpetual change in their environment. Existing decision support methodologies, tools, and frameworks are often difficult to scale up and adapt to changing knowledge. We propose an approach that constructs a decision model from knowledge base according to the changes in the environment. Our work extends the previous work by focusing on how to store knowledge in knowledge base, and how to control the generated Bayesian network model.

This paper presents a new knowledge representation framework to support model construction during runtime from a domain knowledge base. The proposed knowledge representation framework permits the knowledge base designer to specify knowledge in BN fragment instances, and the relationships among BN fragment instances. A BN fragment instance is a sub-model that is used to represent probabilistic knowledge for some part of the domain. The domain knowledge base comprises many sub-models organized in a hierarchy and partition structure. This paper also describes the process for constructing Bayesian network models from the domain knowledge and discusses Bayesian networks combination method in details. The generated Bayesian network can be changed as the situation evolves.

This paper is organized as follows: In Section 2, we describe the proposed knowledge representation scheme briefly. In Section 3, we describe the model construction process. In Section 4, we describe Bayesian networks combination and discuss influence combination methods. In Section 5, we use an example to demonstrate that the generated Bayesian network model may be constantly updated as the situation evolves. Finally, in Section 6, we conclude the paper.

Knowledge Representation

Knowledge experts often consider a related set of variables together. It is reasonable to organize domain knowledge in larger chunks. The ability to represent conceptually meaningful groupings of variables and their interrelationships facilitates both knowledge elicitation and knowledge base maintenance (Mahoney and Laskey, 1996).

In our knowledge representation framework, three classes are defined and their instances are used to represent knowledge and knowledge base structure. The three classes are called BN fragment, Net fragment and Basic fragment.

A BN fragment instance is an instance of BN Fragment class and a sub-model that is used to represent probabilistic knowledge for some part of the domain. Larger situation specific models tend to include these sub-models. Our representation framework takes BN fragment instances as its basic units, which consists of a set of attributes and a specific Bayesian network.

Net fragment and Basic fragment are two classes whose instances are used to represent the relationships among theses BN fragment instances. In a domain knowledge base, these BN fragment instances are organized in a hierarchy and partition structure. Basic fragment instances organize BN fragment instances in a partition structure. A basic fragment instance has a set of BN fragment instances. The set of BN fragment instances represent similar problems under different conditions. When composing a partition, we place each distinguished BN fragment instance into one and only one set. In other words, all BN fragment instances in a basic fragment instance are different and each BN fragment instance belongs to one and only one set. Given the specific condition, only one BN fragment instance is selected from the set of BN fragment instances. Given different conditions, the same BN fragment instance may be selected. The partition structure allows the knowledge engineer or expert to compare similar problems and provide accurate information. Net fragment instances organize BN fragment instances in a hierarchy structure. A net fragment instance has a set of net fragment instances or basic fragment instances. The set of net fragment instances or basic fragment instances are at the next level of the hierarchy. The hierarchy structure allows the knowledge engineer or expert to represent the same knowledge in different hierarchy structures according to their preferences.

Given a specific domain, a large set of BN fragment instances and the relationships among theses BN fragment instances are stored in the domain knowledge base. The knowledge engineer or expert encodes BN fragment instances and the relationships among these BN fragment instances in the form of net fragment instances and basic fragment instances.

Model Construction Process

Based on external inputs, the proposed model construction approach retrieves BN fragment instances from the domain knowledge base, attaches evidence to the Bayesian networks in the selected BN fragment instances and combines the Bayesian networks in the selected BN fragment instances.

The steps for model construction are:

1. Map the possible inputs (e.g. events) to hypotheses, conditional information and evidences according to the mapping information, which is recorded in the domain knowledge base.

   The possible external inputs from sensors or user must be mapped to hypotheses, conditional information and evidences. Each hypothesis is triggered by one or multiple events. We use Bayesian network to represent the relationships between these events and hypotheses. In other words, Bayesian network is used to implement the mapping of hypotheses. The mapping for conditional information and the mapping for evidences are direct mappings from the external inputs to conditional information and evidences.

2. Based on hypotheses and conditional information, select BN fragment instances from the domain knowledge base.

   Hypotheses and conditional information are used for BN fragment instances selection. Each hypothesis corresponds to a net fragment instance or a basic fragment instance in the domain knowledge base. Given hypotheses, the corresponding net fragment instances or basic fragment instances can be obtained by retrieving the domain knowledge base. Given the corresponding net fragment instances or basic fragment instances and conditional information, the BN fragment instances can be selected.

3. Given the selected BN fragment instances, attach the evidence to the Bayesian networks in the selected BN fragment instances and combine the Bayesian networks in the selected BN fragment instances.

   After the BN fragment instances are selected, evidences are attached to the corresponding nodes of Bayesian networks in the selected BN fragment instances. Then, the Bayesian networks in the selected BN fragment instances are combined. The combination result is a Bayesian network model. In Section 4, we will discuses Bayesian Networks combination algorithms.

The proposed approach takes the bottom-up construction approach. For a specific domain, the domain knowledge base stores a large number of BN fragment instances. For a specific problem, a Bayesian network model is generated based on the domain knowledge base and external inputs. When the external inputs change, the hypotheses, conditional information and evidences may be changed or updated. These changes will affect the selected BN fragment instances and in turn lead to the changes in the generated Bayesian network model. Therefore, the generated Bayesian network model may be constantly updated as the situation evolves.

Bayesian Networks Combination

Izhar Matzkevich and Bruce Abramson (1992, 1993) have already done some research on the combination of Bayesian Network. The purpose of their work is the coordination of information provided by multiple experts. They propose an approach that is the fusion of the contributed networks by multiple experts into a single consensus model. Their approach focuses on the graphical combinations.

Our work is to combine some Bayesian networks, each of which represents knowledge for some parts of the domain into a Bayesian network model. As we know, a set of Bayesian networks are pairwise acyclic and yet combination of them can be cyclic. Therefore, it is not sufficient to simply consider those Bayesian networks to be directed acyclic graphs. In a real domain, if the combination of those Bayesian networks is cyclic, the combination cannot be converted into a Bayesian network by reversing arcs because it is necessary to keep those Bayesian networks consistent after the combination. In this case, the knowledge base engineer needs to modify those Bayesian networks that are encoded by knowledge base engineer and are stored in knowledge base in advance. Our approach requires evaluating Bayesian networks' suitability for combination before combining them.

Evaluating Bayesian Networks' Suitability for Combination

In each Bayesian network, there are several nodes. The identifier of a node is the node's name. Common node is the node that exists in at least two different Bayesian networks. The common node in different Bayesian networks must have the same states and yet its probabilities in different Bayesian networks may be different.

The combination of Bayesian networks is a graph that includes all nodes and links of Bayesian networks, and the nodes with same name are combined into a common node. Given a combination, a method is needed to check whether the combination is a directed acyclic graph. If there is a loop in the combination, the combination is not a Bayesian network. The following is an algorithm for finding loops in a graph (X, L). X is a set of nodes and L is a set of links. The adjacency set of X_i is the set of nodes directly attainable from X_i, that is $\mathrm{Adj}(X_i) = \{X_j \in X | L_{ij} \in L\}$

Finding loops algorithm:

For each pair of common nodes (X_i and X_t), if there exists a path from X_i to X_t, and a path from X_t to X_i, then a loop has been found in the graph.

Algorithm for finding path from X_i to X_t:

1. Initialization: Path= ø, and Visited =ø.
2. Add X_i to Visited and set Path = $\{X_i\}$, then take X_k=X_i and Previous = X_i.
3. Iteration: If there exists $X_t \in$ Adj (X_k), then add X_t to Path and stop (a path has been found); otherwise, go to Step 4.
4. If all nodes in Adj(X_k) have already been visited, or Adj(X_k) = ø, go to Step 6; otherwise, go to Step 5.
5. Forward step: Choose some $X_r \in$ Adj(X_k), such that $X_r \notin$ Visited. Set Previous = X_k, add X_r to both Path and Visited, let $X_k = X_r$, and go to Step 3.
6. Backward step: Remove X_k from Path. If $X_k = X_i$, then stop (there is no path in the graph; otherwise, let $X_k = X_i$, let X_k be the last node in Path, and go to Step 4.

During the mode construction process, if evaluation is failure, then the construction process has to stop. If evaluation is successful, then the next step is the combination of influence.

Influence Combination

Nodes in Bayesian networks already have a probability distribution. Suppose that some node X has a parent X_f in a Bayesian network and a parent X_m in another Bayesian network. So that we have probability distributions $P(X| X_f)$ and $P(X | X_m)$. When the two Bayesian networks are combined, we must merge these two models to get a CPT for $P(X | X_f, X_m)$. There are several ways to do this. Laskey and Mahoney (1997) define several influence combination methods to combine conditional probability distributions. The following review some methods and propose interval probability method.

- Simple-Combination: The most straightforward influence combination method is Simple-Combination, which requires the node X to be the resident in exactly one Bayesian Network containing all its parents. Simple-Combination simply employs the distribution for all parent variables.
- Preset-Combination: In preset-Combination, a probability distribution for the node X is overridden by a distribution defined for a more specific set of parent variables, which are recorded in knowledge base in advance.
- Noisy-OR: Noisy-OR is a method for combining partial influences The noisy-or model (Srinivas, 1993) allows one to compactly specify a conditional distribution when the parents are independent, stochastic causes of the child.
- Interval probability method: The Interval probability method is an approximate influence combination method, in which interval probability distributions (Tessem, 1992; Ha et al, 1998) are used. Using this

method, the probabilities at the combined node are interval rather than point –valued probabilities.

An influence combination method has a set of enabling conditions specifying requirements for applicability of the method. The enabling conditions provide a way for knowledge engineer to specify which combination method should be used for a given node. For example, the selected combination method is Simple-Combination, if node X is a resident in exactly one Bayesian network containing all its parents.

Algorithms

Here, we propose two algorithms for Bayesian networks combination. One is exact algorithm; the other is an approximate algorithm. The above influence combination methods are employed in our combination algorithms.

Algorithm 1:

1. Evaluate Bayesian networks' suitability for combination. If evaluation is failure, the construction process has to stop. The knowledge base engineer needs to modify those Bayesian networks. If evaluation is successful, go to Step 2.
2. Combine the graphs of Bayesian networks. The combination result is an acyclic graph that includes all nodes and links of Bayesian networks, and the nodes with same name are combined into a common node.
3. Assign probability distribution for each node. If the node is not a common node, use the original probability distribution. If the node is a common node, use the following influence combination method. Let X be a common node. The selected influence combination method for X is
    a. Simple-Combination if node X is a resident in exactly one Bayesian network containing all its parents.
    b. Noisy-OR if Noisy-OR combination method is specified for the node X.
    c. Preset-Combination if the above two conditions cannot be satisfied and there is a preset distribution in the knowledge base.
    d. Manual input if all the above conditions cannot be satisfied.

Algorithm 1 is an exact algorithm. However, the combination algorithm is not flexible. It requires knowledge engineer to store a lot of distributions in knowledge base in advance. For a specific common node, if conditions a, b and c cannot be satisfied, the combination process will stop and wait for knowledge engineer or user to input probability distribution.

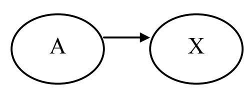
Algorithm 2:

1. Evaluate Bayesian networks' suitability for combination. If evaluation is failure, the construction process has to stop. The knowledge base engineer needs to modify those Bayesian networks. If evaluation is successful, go to Step 2.
2. Combine the graphs of Bayesian networks. The combination result is an acyclic graph that includes all nodes and links of Bayesian networks and the nodes with same name are combined into a common node.
3. Assign probability distribution for each node. If the node is not a common node, use the original probability distribution. If the node is a common node, use the following method. Let X be a common node. The selected influence combination method for X is
    a. Noisy-OR if Noisy-OR combination method is specified for the node X.
    b. Interval probability method if node X does not satisfy Noisy-OR requirement.

Algorithm 2 is an approximate algorithm. Given a common node, if Noisy-OR combination method is not specified for the node, interval probability method will be used. The probabilities at the combined node are interval. Using the algorithm, the Bayesian network combination process can proceed automatically without human interventions if evaluation is successful. Figure 1 shows an example for Bayesian networks combination using algorithm 2.

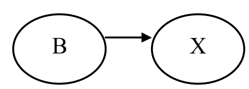
In Figure 1, two Bayesian networks (Figure 1(a) and Figure 1 (b)) are used for combination. First, we evaluate the Bayesian networks' suitability for combination. The verification is successful and there is a common node (X). The node X has a parent A in a Bayesian network and a parent B in another Bayesian network. When the two Bayesian networks are combined, the nodes (A and B) use the original probability distributions while the common node (X) uses interval probability method to merge. Figure 1(c) show the combination result.

We had implemented the two Bayesian network combination algorithms. Users can choose any one according to their preference.



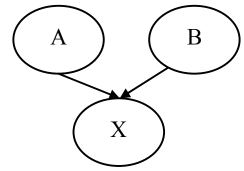
	a1	a2
x1	0.7	0.4
x2	0.3	0.6

a



	b1	b2
x1	0.9	0.5
x2	0.1	0.5

b



	a1		a2	
	b1	b2	b1	b2
x1	[0.7,0.9]	[0.5,0.7]	[0.4,0.9]	[0.4,0.5]
x2	[0.1,0.3]	[0.3,0.5]	[0.1,0.6]	[0.5,0.6]

c

Figure 1 Bayesian Networks Combination

Example

We develop an intelligent decision system that is capable of constructing a decision model during runtime from a domain knowledge base. We will use an example from domain-littoral threats assessment to demonstrate that the generated Bayesian network model may be constantly updated as the situation evolves. In this example, the domain knowledge base of littoral threat assessment stores a lot of BN fragment instances, net fragment instances, basic fragment instances and mapping information. All instances and mapping information are encoded by the knowledge engineer, expert or user.

Figure 2 shows the mapping information of hypotheses. There are two events and two hypotheses. The two events are “foreign officer come by ship” and “suspicious boat”. Two hypotheses are hypothesis H1 –“attack on sector A” and hypothesis H2- “attack on sector A”. The Bayesian network shown in Figure 2 is used to represent the relationships between these events and hypotheses. In this example, assuming that, the hypothesis is triggered when the probability of a hypothesis being true is greater than 0.7. For the event “foreign officer come by ship”, hypothesis H1 is triggered when the probability of the hypothesis H1 is greater than 0.7 after assigning the event as true.

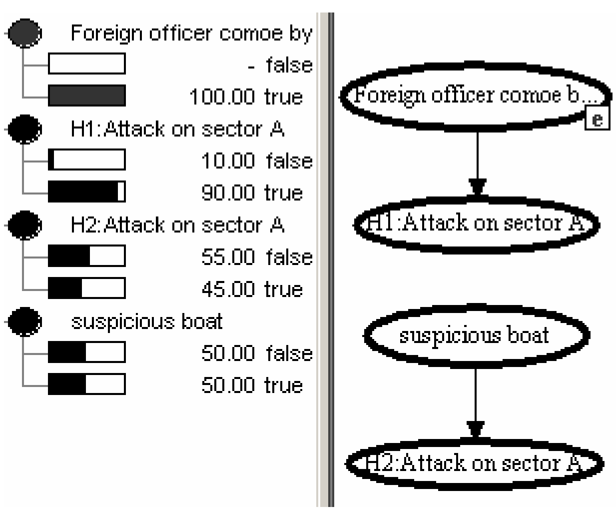


Figure 2 Hypotheses Mapping

In this example, there are two kinds of conditional information, namely “aggressive” and “conservative”.

Given the hypothesis – H1, and the conditional information – aggressive, using the algorithm 1, the system generates a Bayesian network model in Figure 3 based on the domain knowledge base and external inputs (H1 and aggressive).

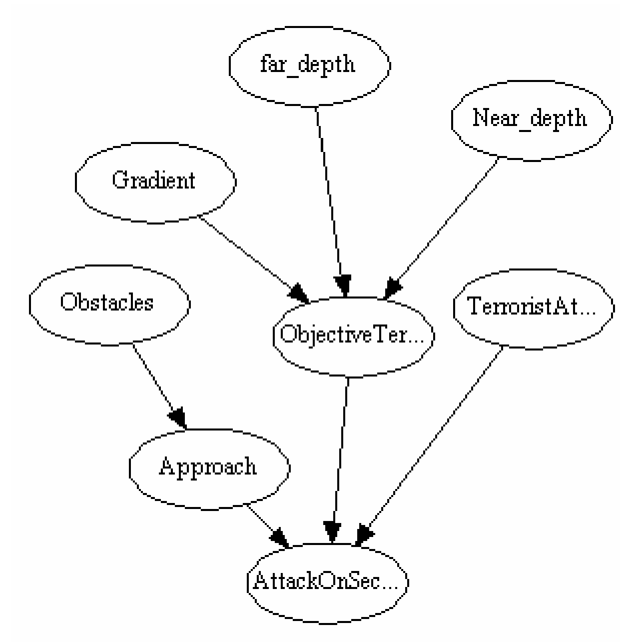


Figure 3 Bayesian Network model generated based on the hypothesis – H1 and the conditional information – aggressive

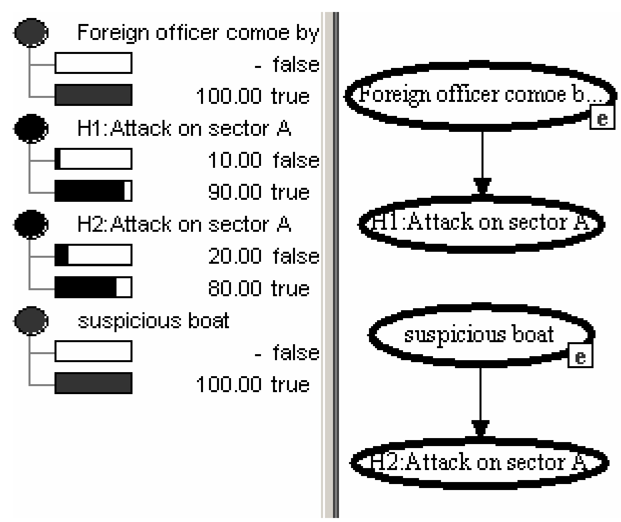


Figure 4 Hypotheses Mapping

Figure 4 shows the mapping information of hypotheses after a new event “suspicious boat” arrives. The probability of the hypothesis H2 being true is 0.8 which is greater than 0.7 after assigning the event as true, therefore hypothesis H2 is also triggered. In other words, when the two events are true, the two hypotheses are triggered.

Given the hypotheses - H1 and H2, and the conditional information - aggressive, using the algorithm 1, the system generates a new Bayesian network model in Figure 5 according to the new situation.

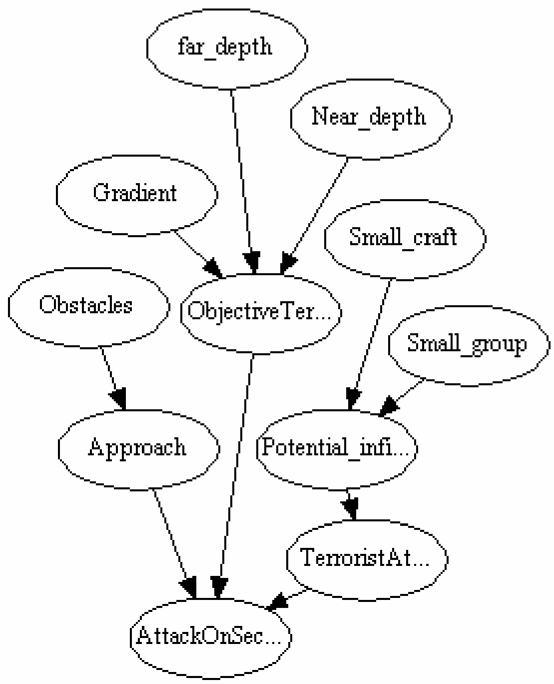


Figure 5 Bayesian Network model generated based on the hypotheses – H1 and H2 and the conditional information – aggressive

In this example, when a new event- "suspicious boat" arrives, the Bayesian network model will be generated according to the new situation. Therefore, the generated Bayesian network model can be dynamically updated. The example from domain-littoral threats assessment shows that our approach is capable of constructing situation-specific Bayesian network models during runtime from a domain knowledge base.

Conclusion

In recent years, many knowledge representation schemes have been developed for the construction of probabilistic and decision models. We propose a new knowledge representation framework to support model construction during runtime from a domain knowledge base. Our work extends the previous work by focusing on how to store knowledge in knowledge base, and how to control the generated Bayesian network model. The proposed knowledge representation framework takes BN fragment instance as its basic unit and organizes these BN fragment instances in a hierarchy and partition structure. The knowledge structure facilitates knowledge representation and acquisition. This paper also describes the model construction process, one of the key steps of which is Bayesian networks combination. The paper presents Bayesian networks combination method and discusses influence combination method. An example is used to demonstrate that the generated Bayesian network can be changed as the situation evolves.

References

Ha, V. A.; and Haddawy, P. 1998. Geometric Foundations for Interval-based Probabilities. *Annals of Mathematics and Artificial Intelligence* 24(1-4): 1—21.

Haddawy, P. 1994. Generating Bayesian Networks from Probability Logic Knowledge Bases. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, 262-269, Morgan Kaufmann.

Laskey, K. B.; and Mahoney, S. M. 1997. Network Fragments: Representing Knowledge for Constructing Probabilistic Models. In *Proceedings of the Thirteen Conference on Uncertainty in Artificial Intelligence,* 334-341, Morgan Kaufmann.

Mahoney, S. M.; and Laskey, K. B. 1996. Network Engineering for Complex Belief Networks, In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, 484-490, Morgan Kaufmann.

Matzkevich, L.; and Abramson, B. 1992. The Topological Fusion of Bayes Net. In *Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence*, 191-198, Morgan Kaufmann.

Matzkevich, L.; and Abramson, B. 1993. Some Complexity Considerations in the Combination of Belief Networks. In *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence*, 152-158, Morgan Kaufmann.

Ngo, L.; and Haddawy, P. 1997. Answering Queries from Context-sensitive Probabilistic Knowledge Bases, *Theoretical Computer Science* 171(1–2): 147–177.

Poole, D. 1993. Probabilistic Horn Abduction and Bayesian Networks. *Artificial Intelligence* 64(1): 81-129.

Srinivas, S. 1993. A Generalization of the Noisy-or Model. In *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence*, 208-218, Morgan Kaufmann.

Tessem, B. 1992. Interval Probability Propagation. The *International Journal of Approximate Reasoning* 7: 95—120.