An Empirical Comparison of Bayesian and Credal Networks for Dependable High-Level Information Fusion

Alexander Karlsson
School of Humanities and Informatics
University of Skövde
Sweden
Email: alexander.karlsson@his.se

Ronnie Johansson
School of Humanities and Informatics
University of Skövde
Sweden
Email: ronnie.johansson@his.se

Sten F. Andler
School of Humanities and Informatics
University of Skövde
Sweden
Email: sten.f.andler@his.se

Abstract—Bayesian networks are often proposed as a method for high-level information fusion. However, a Bayesian network relies on strong assumptions about the underlying probabilities. In many cases it is not realistic to require such precise probability assessments. We show that there exists a significant set of problems where credal networks outperform Bayesian networks, thus enabling more dependable decision making for this type of problems. A credal network is a graphical probabilistic method that utilizes sets of probability distributions, e.g., interval probabilities, for representation of belief. Such a representation allows one to properly express epistemic uncertainty, i.e., uncertainty that can be reduced if more information becomes available. Since reducing uncertainty has been proposed as one of the main goals of information fusion, the ability to represent epistemic uncertainty becomes an important aspect in all fusion applications.

Keywords: High-level information fusion, credal networks, Bayesian networks, dependability, epistemic uncertainty, imprecise probability

I. INTRODUCTION

One of the main goals in information fusion (IF) is to reduce uncertainty by utilizing multiple sources of information [1], and thereby enable better decision making. In fact, this type of uncertainty has been acknowledged in other research fields as epistemic uncertainty [2], i.e., uncertainty that can be reduced if more information becomes available. A common argument that proponents of the set of theories included under the term imprecise probability convey, is that precise Bayesian theory cannot adequately represent epistemic uncertainty [3]. This can be demonstrated with such a simple example as tossing a coin. In the precise Bayesian school, assuming no prior knowledge about the coin, the same probability of “Head” can be adopted as the belief before any information is available, as a prior, as well as later when a huge amount of information is available, as a posterior. In imprecise probability [3], this case amounts to representing the prior of “Head” as a probability interval [0, 1], and a posterior with a smaller interval. The idea is that when a large amount of information is available, then the interval converges into a point estimate, i.e., the degree of imprecision reflects the amount of epistemic uncertainty.

The difference of the two approaches can be substantial in a decision context. In precise probability, a common automated decision-making strategy is to decide for the alternative that maximizes the expected utility with respect to a utility function and a probability distribution [4]. This is an approach that does not take into consideration the amount of information that the distribution is based on, i.e., it is possible to decide for an alternative based on prior ignorance (no information is available) and not even be aware of it. In imprecise probability, prior ignorance reveals itself through the degree of imprecision in both probabilities and expected utilities.

However, since imprecise probability adds considerable computational complexity to algorithms, a question one can pose is how much the additional information about imprecision affects automated decision making. In other words, does imprecision in probabilities provide a better decision support for some set of problems? We are convinced that the main criterion when measuring fusion performance of different uncertainty management methods (UMMs) is to evaluate with respect to decisions. From an automated-decision perspective this amounts to comparison of different methods for finding the best alternative given a set of available information. This comparative aspect of evaluating different UMMs with respect to decisions is an important research topic that has been overlooked in the fusion community. We report on initial work of empirical comparison of the precise probability approach, Bayesian networks (BNs) [5], to an imprecise probability approach, credal networks (CNs) [6], [7], from an automated decision perspective. We start with two elementary research questions, namely, does there exist problems where CNs outperform BNs, and if so, how frequent are such problems? Our hypothesis is that there is a significant set of such problems and that the value to the decision makers, of using a CN in comparison with a BN, is higher when information is scarce (cf. [8]).

The article is organized as follows: in Sect. II, we present our view of high-level information fusion, and depict the theory of BNs and CNs; Sect. III describes the experiments, and results; in Sect. IV and V, we discuss the results and
II. BACKGROUND

High-level information fusion (HLIF) is a research area that has gained less attention historically than low-level information fusion. Most papers addressing HLIF involve graphical models, mainly BNs. We here present our view of HLIF from a perspective of uncertainty management. We also depict the main theory of BNs and CNs, which is later used when describing the experiment design.

A. High-Level Information Fusion

With HLIF we refer to Level 2 – Situation Assessment, and Level 3 – Impact Assessment, in the revised JDL model [9]:

- **Level 2 – Situation Assessment:** estimation and prediction of relations among entities
- **Level 3 – Impact Assessment:** estimation and prediction of effects of actions on situations

As can be seen from the revised JDL model, the main concern in Level 2 – Situation Assessment is relations among entities. Assume that the state of an entity \( i \) is denoted by \( X_i \) with possible values in the set \( \{x_{i1}, \ldots, x_{ik}\} \) and that we use the notation \( X = X_1 \times \ldots \times X_n, x \in X \), i.e., \( X \) is the Cartesian product among entities and \( x \) is a specific element of this product, which we refer to as a situation. A relation can be defined intentionally by a predicate \( \Phi \) that decides which n-tuples belong to the relation [10], thus by using our notation, we can state that the main issue in Situation Assessment is to find \( R \) defined as:

\[
R = \{ x : \Phi(x|\xi) = \text{True} \},
\]

(1)

where \( \xi \) is a set of available information concerning the process (e.g., sensor data). Assume that the entities reflect some process of interest, i.e., each n-tuple in \( X \) is a possible description for the current state of the process. Since we are most often unable to observe all aspects of the process, we cannot simply assign a truth value to \( \Phi(x|\xi) \), \( x \in X \). What we need is a method for assessing a belief measure that expresses the strength of our belief of \( \Phi(x|\xi) \) being true or not for a specific situation \( x \). This representation schema can also be utilized in Level 3 – Impact Assessment, but here we are also interested in how different alternatives in a set \( A = \{a_1, \ldots, a_m\} \) affect the current situation, i.e., assessment of a belief measure for different possible future situations, given an alternative, and a belief measure for the current situation [11].

Research on HLIF has been relatively scarce in comparison with low-level IF, and most of it concerns BNs (e.g., [12], [13]). Furthermore, research concerning HLIF has rarely addressed dependability issues (cf. [14]). If we define a HLIF service as providing decision support for situations, we regard dependability as consisting of three main concepts: reliability, robustness, and stability (for more detail, see [11]). Reliability is a correctness criterion where each involved belief measure should reflect the information of which it is based on. A HLIF service fulfills robustness if the service satisfies reliability in exceptional cases (e.g., lack of information). Robustness with respect to dependability of HLIF is a more generic concept in comparison to robust Bayesian analysis [15], [16], where one mainly think of robustness as sensitivity analysis. Lastly, such a service fulfills stability if small variation in input does not cause a major change of the decision.

B. Bayesian Networks

Let \( X \) be a multivariate random-variable over situations \( x \in X \). A BN [17, Chapter 14] is a representation that allows for easier computation and specification of a joint probability distribution \( p(X) \) for a problem domain involving entities \( \{X_1, \ldots, X_n\} \) (here interpreted as random variables). A BN consists of [5]:

1. A directed acyclic graph with \( \{X_1, \ldots, X_n\} \) as nodes with arcs that capture conditional independence relationships.
2. A set of corresponding conditional probabilities, listed in tables usually referred to as conditional probability tables (CPTs), denoted by \( p(X_i|pa(X_i)) \), where \( pa(X_i) \) is the instantiation of node \( X_i \)'s parents

A certain node, \( X_i \), in a BN is said to be conditional independent of its non-descendants (\( B \) is a descendant of \( A \) if there is a directed path from \( A \) to \( B \) ) given its parents \( pa(X_i) \). When specifying the graph structure, it is common to think in terms of direct influence [17, Section 14.2]. A CPT depicts a probability distribution for a node, given that values for its parents are known. The following formula can be utilized in order to calculate the joint probability distribution based on the graph structure and the CPTs:

\[
p(X) = \prod_{i=1}^{n} p(X_i|pa(X_i))
\]

(2)

C. Credal Networks

A CN [6], [7], [18] is a graphical probabilistic approach, similar to a BN, but where one utilizes imprecision in probabilities. While BNs stipulate precise probabilities, i.e., a single probability distribution, CNs allow one to specify sets of such distributions, thus, a CN can be depicted as a set of BNs for a given graph structure [19]. These sets of distributions, referred to as credal sets [7], are most often assumed to be convex (an assumption that is used throughout this article). As an example, probability intervals for events can be interpreted as constraints for a credal set. Credal sets belong to a family of belief theories usually referred to as imprecise probability [20]. There exist many sources of imprecision in probabilities such as: lack of information, conflicting information (inconsistent information), and conflicting beliefs (e.g., conflict amongst a group of domain experts), to name a few [3]. A credal set can be obtained as a model of belief by thinking in terms of sensitivity analysis as is done in the robust Bayesian school [15], [16], or by subjective domain expert statements, e.g., “A is more probably than B” [20] that then become constraints on a feasible region of probabilities. A credal set can also be obtained by using a model that reflects epistemic uncertainty in a direct way, i.e., the model reflects the amount of information
it is based on. This latter approach to construct credal sets is the one utilized in this article, where the *imprecise Beta model* [3, Section 5.3.2] is used for specifying credal sets involved in our experiment.

Let us denote a credal set by $\mathcal{P}_X$, which contains distributions in the form $p(X_1), \mathcal{P}_{X_1|x_2}$ for distributions in the conditional form $p(X_1|x_2)$, and $\mathcal{P}_{X_1,X_2}$ for joint distributions $p(X_1,X_2)$. Let $\text{ext}(\mathcal{P})$ define the set of extreme points of $\mathcal{P}$, i.e., distributions that cannot be expressed as a convex combination of other distributions in the set. One of the controversies when it comes to imprecise probability is how one should interpret independence concepts between variables [21]. We adopt the most common interpretation of independence, referred to as *strong independence* [7], [21], which states that a variable $X_1$ is independent of $X_2$ if the following holds:

$$\mathcal{P}_{X_1,X_2} = CH\left\{ p_i p_j : p_i \in \mathcal{P}_{X_1}, p_j \in \mathcal{P}_{X_2} \right\}, \quad (3)$$

where $CH$ is the convex-hull operator. Similarly, a variable $X_1$ is conditional independent of $X_2$ given $X_3$ if $\forall x_3$:

$$\mathcal{P}_{X_1,X_2|x_3} = CH\left\{ p_i p_j : p_i \in \mathcal{P}_{X_1|x_3}, p_j \in \mathcal{P}_{X_2|x_3} \right\} \quad (4)$$

A CN can be depicted as a collection of separately specified credal sets $\mathcal{P}_{X_1|pa(X_1)}$ [19] of distributions in the form $p(X_1|pa(X_1))$. Since we have stated that a credal set is convex, it suffices to use its extreme points as a representation for the entire set. The largest joint credal set based on strong independence, referred to as the *strong extension*, is defined by [7], [18], [19]:

$$\mathcal{P}_X = CH\left\{ \prod_{i=1}^n \text{ext}\left( \mathcal{P}_{X_i|pa(X_i)} \right) \right\}, \quad (5)$$

where the product operator between any two credal sets, $\mathcal{P}_{X_1}$ and $\mathcal{P}_{X_2}$, is defined as:

$$\mathcal{P}_{X_1} \mathcal{P}_{X_2} = \{ p_i p_j : p_i \in \mathcal{P}_{X_1}, p_j \in \mathcal{P}_{X_2} \} \quad (6)$$

Upper and lower bounds for the probability of a situation, can be obtained by: $\underline{p}(X) = \inf_{p \in \mathcal{P}_X} p(X)$ and $\bar{p}(X) = \sup_{p \in \mathcal{P}_X} p(X)$. Given that a utility function is available for each situation $x \in X$ in conjunction with every possible alternative $a \in \mathcal{A}$, i.e., $u : X \times \mathcal{A} \rightarrow \mathbb{R}$, it is possible to calculate upper and lower bounds for the expected utility of different alternatives in the following way [7]:

$$\bar{E}_{u,p}(a) = \inf_{p \in \mathcal{P}_X} E_{u,p}(a) \quad (7)$$

$$\underline{E}_{u,p}(a) = \sup_{p \in \mathcal{P}_X} E_{u,p}(a) \quad (8)$$

$$E_{u,p}(a) = \sum_{x \in X} p(X = x)u(x,a) \quad (9)$$

From a decision maker’s point of view, upper and lower bounds on expectations induce a partial ordering on alternatives, i.e., there may exist pair of alternatives that are not comparable with each other [22].

### III. Credal Networks Vs. Bayesian Networks

We base our design of experiments on a method that Aughenbaugh and Paredis [8] have introduced for evaluating the value of imprecise probability in engineering design. Assume that we want to have a competition between two methods encapsulated by an *agent* paradigm. The following information is available to the agents:

- A graphical model
- Samples from the CPTs in the graphical model
- A utility function $u : X \times \mathcal{A} \rightarrow \mathbb{R}$

Since the agents only get samples from the probabilities in the graphical model, they can at best have beliefs (here used as a generic term that is not associated with a particular UMM) about the true probabilities. The *goal* of each agent in the competition is to win, something that is defined as deciding on an alternative that produces the highest value of true expected utility. Assume further that there is an agent that acts as a *supervisor* for the competition and that besides having access to the above information also knows the true distributions. Consequently, the supervisor knows about the true expected value for each alternative in $\mathcal{A}$, thus also which alternative that is best. Hence, the supervisor can be considered to be the optimal agent with respect to the competition, and should therefore be used as a basis for how well the other agents perform.

The basic flow of the competition as presented by Aughenbaugh and Paredis is: given the information specified above, the agents make a decision about the best alternative in $\mathcal{A}$, then, since the supervisor knows the true expected utility of the selected alternatives, it is able to compare the true expected value of the agents’ decisions. An agent wins over the other if it decides for an alternative that generates a higher expected utility. By performing this a large number of times, it is possible to get a good estimate of the probability for the event that a specific agent wins.

#### A. Experiment Design

Assume that the graphical model has the simple structure as in Fig. 1, and that all variables are binary. Assume further that the agents only have to decide for two alternatives, i.e., $\mathcal{A} = \{a_1, a_2\}$. The first step for each of the agents is to find an appropriate estimate of the distributions involved in the graph, given that a number of samples from each of the distributions are available. We note that since each probability distribution spans only two possible states, it is appropriate to choose the *Binomial distribution* [4] as the sample distribution:

$$\text{Bin}(\#x|\theta, N) = \binom{N}{\#x} \theta^{\#x} (1 - \theta)^{N - \#x}, \quad (10)$$

where $\theta$ is the probability of observing a “success” $x$, “$\# = \text{number of}$”, and $N$ denotes the number of observations. Each agent needs to find what it think is the best possible estimate for the parameter $\theta$ in each of the distributions in the graph. These estimates are then used to calculate the joint probability distribution and credal set over $X$, which in turn is used for
determining the expected utility of the two actions in \( \mathcal{A} \). We model three agents, one based on a BN, agent \( B \), and two based on a CN, agents \( C_1 \) and \( C_2 \).

1) Agent \( B \) - Bayesian Network: A reasonable approach for the agent that utilizes a BN, with consideration to that it has accepted precise Bayesian theory, is to adopt a uniform Beta distribution, \( \text{Bet}(\theta|1,1) \) as the prior [4]. The posterior result is a new Beta distribution, \( \text{Bet}(\theta|\#x + 1, \#\bar{x} + 1) \), where \( x \) denotes a “success”, and \( \bar{x} \) denotes a “failure”. The following expected value of \( \theta \) (with respect to the posterior) is used by agent \( B \), given that it has access to \( N \) samples [5]:

\[
\frac{\#x + 1}{N + 2}
\]

Now, since agent \( B \) knows how to find an estimate for each of the distributions in the graph, it can calculate the joint probability distribution over \( X \) by utilizing Eq. (2). The agent then utilizes this distribution in conjunction with the specified utility function in order to choose the alternative that maximizes the expected utility [4, Section 2.5]:

\[
a_B^* = \arg \sup_{a \in \mathcal{A}} E_{u,p}(a)
\]

2) Agent \( C_1 \) - Credal Network: Agent \( C_1 \) acknowledges that the available information should be reflected by the degree of imprecision in probabilities, i.e., that epistemic uncertainty should be represented, and it therefore adopts the imprecise Dirichlet model [23], which in the case of two possible outcomes reduces to an imprecise Beta model [3, Section 5.3.2]. Instead of assessing a single prior, as was necessary by agent \( B \), agent \( C_1 \) utilizes a set of priors that expresses as much imprecision as possible regarding the probability of event \( x \) ("success"), i.e., \( \theta \in [0,1] \), reflecting that no information is available about the parameter, and hence a maximized amount of epistemic uncertainty. The following interval of expected values for \( \theta \) is utilized by agent \( C_1 \), given \( N \) observations [3, 18, 23]:

\[
\left[ \frac{\#x}{N + s}, \frac{\#x + s}{N + s} \right],
\]

where \( s \) is a learning parameter that determines how fast the interval converges towards a point estimate. A high value on \( s \) implies slower convergence and the other way around.

Notice that when no observations have been made, i.e., \( N = 0 \), the expression above reduces to the interval \([0,1]\), i.e., no information is available, and we therefore have a maximum amount of epistemic uncertainty. We use \( s = 2 \), since this choice encompasses some commonly used Beta priors (see further [23, Section 2.5]). By utilizing Eq. (13), upper and lower bounds for the probability distributions involved in the graphical model can be obtained, which in this case, since \([\Omega_{X_1}] = [\Omega_{X_2}] = [\Omega_{X_3}] = 2\), is equivalent to the distributions’ extreme points \( \text{ext}(\mathcal{P}_{X_1}), \text{ext}(\mathcal{P}_{X_2|x_1}), \text{ext}(\mathcal{P}_{X_2|x_2}) \), \( \text{ext}(\mathcal{P}_{X_3|x_2}) \), and \( \text{ext}(\mathcal{P}_{X_3|x_2}) \).

The decision strategy adopted by agent \( C_1 \) is not as straightforward as in the case of agent \( B \) due to that \( \mathcal{P}_X \) only induces upper and lower bounds of expected utilities for alternatives, which can be overlapping. Agent \( C_1 \) uses a decision strategy called \( \Gamma \)-maximin [24, Section 1.5], a strategy also adopted by Aughenbaugh and Paredis [8]. The main idea in \( \Gamma \)-maximin is that the minimum utility over distributions should be maximized over the alternatives, i.e.:

\[
a_{C_1}^* = \arg \sup_{a \in \mathcal{A}} \left\{ \inf_{p \in \mathcal{P}_X} E_{u,p}(a) \right\}
\]

3) Agent \( C_2 \) - Credal Network: Agent \( C_2 \) uses the same approach as agent \( C_1 \) when it comes to estimating the probabilities, but a somewhat different strategy when it comes to deciding for an alternative. While agent \( C_1 \) utilized a utility function in order to decide for the best alternative, agent \( C_2 \) exploits a two step strategy; it first picks a “good” representative distribution from the credal set, and then uses this distribution for calculating the expected utility in order to decide for an alternative. The agent implements this strategy by choosing the distribution that maximizes the entropy [25–27], i.e., it picks the distribution that represents the highest degree of uncertainty.

Since the number of possible situations is \([\Omega_X] = 2^3 = 8\), we can represent each of the extreme points in \( \text{ext}(\mathcal{P}_X) \) by a probability vector \( v \in \mathbb{R}^8 \). In fact, since \( v \) constitutes a probability distribution we can represent such vectors by utilizing the first seven of its components and the fact that \( v_8 = 1 - \sum_{1 \leq i \leq 7} v_i \), where \( v_i \) denotes the \( i \)-th component of \( v \). Thus, we can transform the complete problem into \( \mathbb{R}^7 \). Assume that agent \( C_2 \) wants to find the maximum entropy distribution \( v^* \in \mathcal{P}_X \) and that we formulate the optimization problem in \( \mathbb{R}^7 \). The extreme points, \( \text{ext}(\mathcal{P}_X) \), define a polytope, which can be expressed as a set of halfspaces, i.e., facets of the polytope. By utilizing the Quickhull algorithm [28], we can find this set of facets, which constitutes linear constraints for the optimization problem. We formulate the optimization problem as a minimization problem (by negating the entropy function) in the following way:

\[
\text{minimize } \sum_{1 \leq i \leq 7} v_i \ln(v_i) + \left( 1 - \sum_{1 \leq i \leq 7} v_i \right) \ln \left( 1 - \sum_{1 \leq i \leq 7} v_i \right)
\]

subject to:

\[
Av \leq b,
\]

where \( v \in \mathbb{R}^7 \), \( A \) is an \( n \times 7 \) matrix and \( b \) an \( n \) vector, where \( n \) depends on the number of facets from the output of Quickhull. The objective function is convex, and the constraints are linear, i.e., the constraints are convex, thus, this problem is a convex optimization problem, which implies that every local minimum
is a global minimum. In fact, the objective function is strictly convex (see Appendix for a proof sketch), thus, the solution $v^*$ is also unique [29, Proposition 4.11].

As the last step, agent $C_2$ decides for the alternative that maximizes the expected utility with respect to $v^*$:

$$a^*_{C_2} = \arg\sup_{a \in A} E_{u,v^*}(a)$$  \hspace{1cm} (16)

4) Agent $S$ – The Supervisor: The supervisor has access to the true probabilities, thus, it always knows the best decision and its true expected utility. Assume that the supervisor collects the decisions made by the agents and then calculates the true expected value for the chosen alternatives; if $E_{u,v}(a^*_{C_1}) > E_{u,v}(a^*_{C_2})$ ($C_1$ refers to $C_1$ or $C_2$) then agent $C_1$ wins, $E_{u,v}(a^*_{C_1}) = E_{u,v}(a^*_{C_2})$ represent a draw, and otherwise agent $C_2$ wins. For a specific problem instance where a specific number of samples is available from each of the distributions in the graphical mode, a competition can be performed by two agents where the supervisor decides for the winner. By repeating the competition a large number of times, it is possible to get a good estimate of the probability for the event that a certain agent is winning that problem instance.

B. Implementation detail

The experiment was implemented using python 2.5 with Mersenne Twister [30] as the pseudo random number generator (with a period of $2^{19937} - 1$ [31]).

1) Agent $C_1$: $E_{u,v}(a)$ in Eq. (14) is a linear expression, thus, we know that an optimal solution to $\inf_{p \in \mathcal{P}_X} E_{u,v}(a)$ in Eq. (14), exists among $\text{ext} (\mathcal{P}_X)$ [29, Theorem 4.12]. Therefore it suffices to calculate Eq. (14) without implementing the convex hull operation for $\mathcal{P}_X$ in Eq. (5).

2) Agent $C_2$: Qhull (“qhull Qx s n”) [28] was used in order to find the facets of the polytope (defined by the extreme points in $\text{ext}(\mathcal{P}_X)$). The convex optimization problem was solved with the convex optimization package CVXOPT [32] (with absolute accuracy, feasibility tolerance set to $10^{-7}$, and relative accuracy set to $10^{-6}$), which is based on a primal-dual interior-point method [33, Chapter 19]. The size of the polytope decreases with an increased number of observations, which can result in singularities and precision problems when solving the optimization problem. Our simulation program accounts for such problems by approximating the convex hull, if necessary. We first use “qhull Qx s n”, and if the optimization problem cannot be solved, we approximate the convex hull with “qhull Qx W1e-4 CX s n”, where “1e-4” is the minimum required distance between a point and a facet in order to consider the point to be “outside” the facet. The option “CX” causes qhull to merge facets and hyperplanes if their center radius is within “x”. We initiate the approximation by letting “x=1e-6” and if a solution to the optimization problem cannot be found, we increase “x” until a maximum of “x=1e-4” has been reached. Due to the convex hull operation and the convex optimization problem, this is a considerably more computationally complex agent in comparison to agent $C_1$.

C. Experiment I – Existence

In this experiment our aim is to find problem instances where an agent based on a CN clearly outperforms an agent based on a BN. We implement a random search in the space of problem instances, i.e., the space of possible combinations of utility functions and distributions:

$$\mathbb{U} \times \mathbb{F}_1 \times \ldots \times \mathbb{F}_5$$

$$\mathbb{U} = \{ u | u : X \times A \rightarrow \{-10, \ldots, 10\} \}$$

$$\mathbb{F}_1 = \{ p | p : X_1 \rightarrow [0,1], p(x_1) + p(\bar{x}_1) = 1 \}$$

$$\mathbb{F}_2 = \{ p_{x_1} | p_{x_1} : X_2 \rightarrow [0,1], p_{x_1}(x_2) + p_{x_1}(\bar{x}_2) = 1 \}$$

$$\mathbb{F}_3 = \{ p_{x_2} | p_{x_2} : X_2 \rightarrow [0,1], p_{x_2}(x_2) + p_{x_2}(\bar{x}_2) = 1 \}$$

$$\mathbb{F}_4 = \{ p_{x_3} | p_{x_3} : X_3 \rightarrow [0,1], p_{x_3}(x_3) + p_{x_3}(\bar{x}_3) = 1 \}$$

$$\mathbb{F}_5 = \{ p_{x_5} | p_{x_5} : X_3 \rightarrow [0,1], p_{x_5}(x_3) + p_{x_5}(\bar{x}_3) = 1 \}$$

The result of these simulations is seen in Figs. 2 and 3, where the estimated probabilities of “CN win”, “a draw”, “BN win” are plotted with a 95% confidence interval. It is seen that the value of using an agent based on a CN is clearly highest at small sample sizes $N$, i.e., when information is scarce (or equivalently, when epistemic uncertainty is high). Note that for these two problem instances, the estimated probability of the event that agent $B$ is winning is zero or close to zero.

<table>
<thead>
<tr>
<th>Table I</th>
<th>PROBABILITY DISTRIBUTION $p(X_1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p(x_1)$</td>
<td>$C_1$ vs. $B$</td>
</tr>
<tr>
<td>0.05</td>
<td>0.70</td>
</tr>
<tr>
<td>0.95</td>
<td>0.30</td>
</tr>
</tbody>
</table>

| Table II | PROBABILITY DISTRIBUTION $p(X_2 | X_1)$ |
|---------|-------------------------------------|
| $p(x_2 | x_1)$ | $C_1$ vs. $B$ | $C_2$ vs. $B$ |
| 0.38 | 0.66 | |
| 0.42 | 0.34 | |
| 0.30 | 0.09 | |
| 0.70 | 0.91 | |

| Table III | PROBABILITY DISTRIBUTION $p(X_3 | X_2)$ |
|----------|-------------------------------------|
| $p(x_3 | x_2)$ | $C_1$ vs. $B$ | $C_2$ vs. $B$ |
| 0.45 | 0.75 | |
| 0.55 | 0.25 | |
| 0.13 | 0.53 | |
| 0.87 | 0.47 | |
D. Experiment 2 – Frequency

Now that we know that there exist cases where a CN approach outperforms a BN, a question of interest becomes how frequent such problem instances are? In order to get some idea of the answer, we draw a problem instance at random, let the agents compete against each other 1000 times, and count the occurrences of the different outcomes. We repeat this procedure 100 times, i.e., we draw 100 problems at random and let the agents compete 1000 times for each such problem. In order to decide a winner for a specific problem, we implement a sign test [34, Section 17-1] at a 5% significance level, where the null hypothesis is “draw”, i.e., the probability of winning is 0.5 for both of the agents. The sign test counts the number of positive and negative valued results of \( E_{u,p}(a^*_1) - E_{u,p}(a^*_2) \) for each problem instance, and test for significance. Thus, there can be 1000 signs or less, since zero differences reduces the sample size in the sign test (we use exact critical values for the sign test when \( l \leq 100 \), where \( l \) is the number of signs, but utilize the approximation \((l-1)/2 - 0.98\sqrt{l+1}\), for \( 100 < l \leq 1000 \) [34, Table A-10a], which actually results in a sign test that is somewhat stronger than its exact counterpart).

In Figs. 4 and 5, it can be seen that the set of problems where agents \( C_1 \) and \( C_2 \) wins is approximately of equal size to the set of problems where agent \( B \) wins. There are quite many instances where the result becomes a “draw”, which is really not a surprise since \( |A| = 2 \). We observe that when more information becomes available, i.e., the sample size increases, the number of instances where a CN and BN are equally good increases, i.e., the number of competitions where the outcome becomes a draw, increases. This is quite natural since the imprecise Beta models utilized by agents \( C_1 \) and \( C_2 \), converges towards the relative frequency of “success”, which is also the case for the Beta distribution utilized by agent \( B \).

IV. DISCUSSION

The results clearly show that there exists a significant set of problems where it is beneficial to represent epistemic uncertainty in the belief measures. The size of the set seems to be approximately the same size as the set of problems where
a BN wins. Let us relate our findings to dependability (see Sect. II-A) with respect to epistemic uncertainty. By using the degree of imprecision as a measure of epistemic uncertainty, we are able to express the amount of information that the belief measure is based on, thus, a CN satisfies reliability. If we consider “lack of information”, closely related to epistemic uncertainty, as exceptional, then a CN also satisfies robustness.

Lastly, assume that we consider the CPTs in a BN as “input”, then, by varying the CPTs to possible values that “agree” with the set of available information (related to epistemic uncertainty), we are able to express the amount of information that the belief measure is based on, thus, a CN satisfies reliability. If we consider “lack of information”, closely related to epistemic uncertainty, as exceptional, then a CN also satisfies robustness.

V. CONCLUSIONS AND FUTURE WORK

We have confirmed our hypothesis that there exists a significant set of problems where it is beneficial to utilize an uncertainty management method that is able to represent epistemic uncertainty. We have also pointed out that there is a strong relation between one of the main goals in information fusion, namely, reducing uncertainty, and epistemic uncertainty, the latter closely coupled with imprecise probability.

It should be stressed that we have only evaluated two possible ways of deciding on an alternative based on a credal set in a CN; there exist a vast amount of different ways such a decision can be made, e.g., E-admissibility [36, Section 5], utilizing the centroid [37] of the joint probability polytope, a Hurwicz criterion (see further [38]), Maximality [3, Section 3.9], pick a distribution at random from the credal set (see further [39]), and Γ-maximax [40]. Furthermore, the involved credal sets themselves can be constructed differently, for example by utilizing a different value on the parameter $s$ (see Sect. III-A2) in the imprecise Beta model.

In our future work, we plan to further investigate different strategies for decision making based on credal sets in CNs and compare these approaches against each other and against BNs. An important objective is to find characteristics of problems where a CN outperforms a BN. A real-world scenario for automated decision making in HLIF, such as precision agriculture [41], will be used for implementing an experiment in a realistic setting.

ACKNOWLEDGEMENT

We wish to thank Professor Henrik Boström for valuable feedback regarding the design of experiments. This work was supported by the Information Fusion Research Program (University of Skövde, Sweden) in partnership with the Swedish Knowledge Foundation under grant 2003/0104 (URL: http://www.infofusion.se).
APPENDIX

We here give a short proof sketch for the claim that the objective function in Eq. (15) is strictly convex. Let:

\[ f(v) = \sum_{i \leq 7} g_i(v) + g_8(v) \]  

(18)

\[ g_i(v) = v_i \ln(v_i), \quad 1 \leq i \leq 7 \]  

(19)

\[ g_8(v) = \left(1 - \sum_{i \leq 7} v_i \right) \ln \left(1 - \sum_{i \leq 7} v_i \right), \]  

(20)

where \( v_i \) denotes the \( i \)-th component of \( v \). We also define \( g_i(v) = 0 \) if \( v_i = 0 \) for \( 1 \leq i \leq 7 \), and \( g_8(v) = 0 \) if \( 1 - \sum_{i \leq 7} v_i = 0 \). Consider the case where \( g_i(v) \neq 0 \), for \( 1 \leq j \leq 8 \), then it can be shown that \( g_j \) are positive definite, i.e., \( x^T \nabla^2 g_j(v)x > 0 \), \( \forall v, x \in \mathbb{R}^7 \), and this is equivalent to strict convexity of each \( g_j \) [29, Theorem 3.41]. We can now use the fact that a positive weighted sum of strictly convex functions is a strictly convex function [42, Section 3.2.1], thus, \( f \) is strictly convex. The set of points that imply \( g_i(v) = 0 \) for some \( i \) does not affect strict convexity of \( f \) since these points are located on the boundary of \( f \)'s domain, where \( \frac{\partial f(v)}{\partial v} \) limits either \( \infty \) or \( -\infty \) for some \( i \).

REFERENCES


