Comparing Stochastic Design Decision Belief Models: Pointwise versus Interval Probabilities

Peter C Matthews
School of Engineering and Computing Sciences
Durham University, UK

Decision support systems can either directly support a product designer or support an agent operating within a multi-agent system (MAS). Stochastic based decision support systems require an underlying belief model that encodes domain knowledge. The underlying supporting belief model has traditionally been a probability distribution function (PDF) which uses pointwise probabilities for all possible outcomes. This can present a challenge during the knowledge elicitation process. To overcome this, it is proposed to test the performance of a credal set belief model. Credal sets (sometimes also referred to as p-boxes) use interval probabilities rather than pointwise probabilities and therefore are more easier to elicit from domain experts. The PDF and credal set belief models are compared using a design domain MAS which is able to learn, and thereby refine, the belief model based on its experience. The outcome of the experiment illustrates that there is no significant difference between the PDF based and credal set based belief models in the performance of the MAS.

Introduction

Modern trends in product development and production has seen a shift from products being fully developed in-house to products being developed in collaboration with ever increasing numbers of external partners. Along with this shift, there has been the need to complement the change in design with a change in the enterprise structure. Where product development previously could be undertaken by a small co-located team generating a design that would be manufactured in-house, it is now more common to see large disperse teams being composed from multiple organ-

Design Computing and Cognition DCC'10. J.S. Gero (ed),
pp. 327–345 ©Springer 2010

isations along with the manufacture of the product being undertaken in a similar multi-site manner [1, 2].

The complexity of modern design and manufacturing introduces a new source of uncertainty [3]. A component of this uncertainty is how well potential partners are able to work together. For example, two organisastions can have a very good (tacit) understanding of their mutual capabilities and constraints and therefore are able to work well together. On the other hand, another pairing of organisations could completely lack this mutual understanding and place unrealistic demands on each other resulting in poor collaboration. It is this ability to identify suitable partners for collaborative work that this paper addresses. There have been efforts on selecting partners through capability profiling [4], however there remain issues on how the individual capability scores are determined. These capability scores are subjective to the capability supplier rather than to the purchaser. There is therefore a need to devise capability metrics that are subjective to the purchaser.

The challenge is how can an agent determine if a collaboration will be successful. To address this, there are a number of assumptions: (1) it is not appropriate to represent capability by a point score, rather a distribution-based representation should be used; (2) for the purposes of simulation based experiments, an abstract measure of success should be used; and (3) the actors within this experiment can be modeled using agents, and the interaction between these can be modeled using a Multi-Agent System (MAS). The experiments will test how well the agents are able to 'learn' suitable probability distribution functions (PDFs) for 'potential successful interaction'.

An important problem in learning PDFs is the uncertainty about the accuracy of the PDF. Specifically, how accurate is the probability that a certain outcome will occur? To address this, two approaches for representing this stochastic information will be compared. The first will be the well known PDF representation, where probabilities for all the outcomes are represented by pointwise values. The second will be the use of credal sets [5] where the probabilities of outcomes are represented by a probability interval. The rationale behind this is that the PDF provides a well known benchmark, with classic learning algorithms and simple summary statistics to be used for decision making. The credal set approach provides greater flexibility and supports a richer representation of an agent's understanding of outcome probabilities. For the credal set approach, similar learning algorithms and decision algorithms will be required. The credal set approach would also enable the knowledge elicitation process to work with interval probabilities rather than pointwise, which are easier to elicit from a domain expert. For example, credal sets have been used to design a pressure vessel with a new material (representing uncertain design conditions). In this pressure vessel design case, it was shown that where the imprecision is large, the credal sets outperformed the classic pointwise probability distribution representation method [6].

The remainder of this paper is structured as follows: The next section introduces the mathematical concepts and representations required to implement both a PDF and credal set belief model. The following section describes the learning algorithm implemented for both belief models. This is followed by the presentation of how these models are implemented in a multi agent system. Next section describes the empirical trial environment. Finally, the results are presented and discussed.

Stochastic representations

Stochastic decision support requires a stochastic domain model. This paper sets out to compare two different approaches for implementing the stochastic domain model. The first will be the well understood probability distribution function (PDF). The PDF will be used as a benchmark to compare the credal set approach. Both of these approaches are expanded on in this section.

Two different approaches to representing stochastic information will be considered. The first is the classical probability distribution function (PDF) representation. Let Ω be the set of all possible outcomes for a random variable, X. A PDF is a function, $f(x) = P(X = x)$ from the set of outcomes, $x \in \Omega$ that maps onto the probability interval $[0, 1]$, subject to the condition that:

$$\sum_{x \in \Omega} f(x) = 1. \tag{1}$$

For the purposes of this paper, only discrete valued outcome spaces will be considered. The arguments remain valid for continuous valued spaces as well, the summations simply need to be replaced by integrals.

Under the PDF approach, the probability of an outcome is *precisely* defined. For example, consider the outcome of a certain collaboration between two agents. The possible outcomes for the collaboration are: fail, poor, fair, excellent. The probability of each outcome can then be presented as: $P(C = \text{fail})$, $P(C = \text{poor})$, $P(C = \text{fair})$, and $P(C = \text{excellent})$. Each of these represents the probability that the collaboration has the stated outcome. While the outcome of the collaboration is unknown, the PDF representation for this case is precise, e.g.:

$$\begin{aligned} P(C = \text{fail}) &= 0.1 \\ P(C = \text{poor}) &= 0.2 \\ P(C = \text{fair}) &= 0.4 \\ P(C = \text{excellent}) &= 0.3 \end{aligned}$$

Given the above values for this system, an observer would expect to see 10% of collaborations fail, 20% of collaborations be poor, 40% of collaborations be fair, and 30% of collaborations be excellent. However when parameterising this case, there might not be such certainty on these values. For example, the expert might not have total confidence in the pointwise values being assigned to both outcomes. A very simple method for overcoming this is to represent the probabilities as intervals rather than as points [5, 7].

The use of intervals for probability outcomes provides a natural extension to the pointwise approach. The length of the interval is related to the confidence, or certainty, of the probability value of an outcome. Specifically, the probability of an outcome is specified as a range of values. Using the above example, it becomes possible to say $P(C = \text{fair}) = [0.3, 0, 5]$: i.e. the probability that the collaboration will be successful lies between 0.3 and 0.5.

Functional distribution representation

In classical probability, the well understood and used method for representing the probability of various outcomes is the probability distribution function (PDF). This can be easily plotted in two dimensions for visualisation using the outcome space as the horizontal axis and the probability of each outcome occurring as the vertical axis. Figure 1 illustrates two PDFs on the same set of axis, illustrating how characteristics (e.g. the distribution mode) can be identified from this representation. For the scope of this work, it is sufficient to consider the discrete case. Let Ω be the set of all possible outcomes for some random variable X (for simplicity of the mathematical notation, it is assumed that Ω is an ordered set). The PDF for this random variable is then defined as:

$$\begin{aligned} f_X(x) &= P(X = x) \\ \text{s.t.} \quad \sum_{x \in \Omega} f_X(x) &= 1 \end{aligned} \tag{2}$$

An equivalent representation is the cumulative distribution function (CDF). This is simply the sum (or integral in the continuous case) of the PDF in Equation 2 along the horizontal axis, and can therefore also be visualised in two dimensions. As the PDF is bounded between 0 and 1 and sums to unity, the CDF is a monotonically increasing function from 0 to 1. The CDF is defined as:

$$F_X(x) = \sum_{i \leq x} f_X(i) \tag{3}$$

Functional representation of interval probabilities

A slightly different approach is required to represent the stochastic information in the interval case. In the interval probability case, the CDF representation can be used to define the full range of possible (pointwise) distribution functions for a given variable. By extending the concept of interval probabilities for each outcome, two CDFs can be defined: firstly the CDF defined by all the lower probability ranges, $\underline{F}(X = x)$ and secondly the CDF defined by all the upper probability ranges, $\overline{F}(X =$

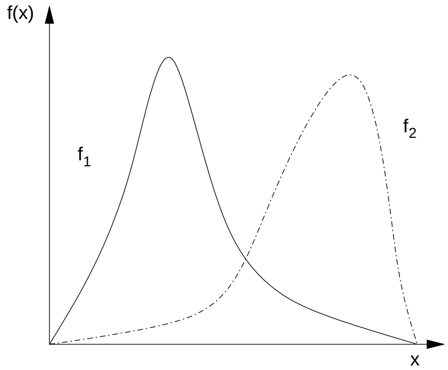

Fig. 1 Two probability distribution functions: f_1 represents a distribution with a low valued mode and f_2 represents a high mode.

x). These two CDFs provide the envelope for all possible CDFs that could represent the distribution function for the given variable.

The p-box [8], or credal set [9], representation for uncertainty is based on an envelope that defines the range of possible cumulative distributions. This is bounded by the maximal ($\overline{F}$) and minimal ($\underline{F}$) values taken by all possible distribution functions for the given uncertainty. This credal set is then formally defined as:

$$\mathscr{M} = \{F : \underline{F}(i) \leq F(i) \leq \overline{F}(i), \forall i \in \Omega\} \tag{4}$$

As the area within the envelope defines the range of possible distribution functions, it follows that the larger this area is the greater variety of distribution functions that exist to represent the variable. Therefore it is possible to numerically define and measure the *uncertainty* about the distribution for the given variable. Numerically, the uncertainty for a given credal set $\mathscr{M}$ will be defined as:

$$\mathrm{Unc}(\mathscr{M}) = \frac{1}{|\Omega|} \sum_{i \in \Omega} \left(\overline{F}(i) - \underline{F}(i)\right) \tag{5}$$

From this equation, it can be seen that where the credal set 'narrows' to the limit, $\mathrm{Unc}(\mathscr{M}) = 0$ and conversely where the credal set contains all possible distribution functions, $\mathrm{Unc}(\mathscr{M}) = 1$.

To illustrate the credal set, consider again the random variable C representing the success of a collaboration between two agents. Now, the probabilities of each outcome is represented by an interval.

$$\begin{aligned} P(C = \text{fail}) &= [0.0, 0.2] \\ P(C = \text{poor}) &= [0.1, 0.3] \\ P(C = \text{fair}) &= [0.3, 0.5] \end{aligned}$$

$$P(C = \text{excellent}) = [0.1, 0.4]$$

These intervals define the range of distribution functions that represent the probabilities for each possible outcome, subject to the probabilities summing to unity. Within this range there are infinitely many possible distribution functions. The credal set has the property that for any two distributions taken from the credal set say F_1 and F_2, all linear combinations of the form $\alpha F_1 + (1-\alpha)F_2$, for $\alpha \in [0,1]$ will also be a member of the credal set.

Agent learning

The agents within this system learn (gain experience) through interaction and observation. The learning is based on a pair of agents initiating a collaboration. This collaboration will have some measurable degree of success, and this evidence is used by the agents to learn about their mutual ability to collaborate. Specifically, the learning process modifies the agent's stochastic information held about the observed variable (in this case the ability to successfully collaborate). If the current stochastic belief is given by F and the observed evidence is given by E, then the updated stochastic belief is determined as follows:

$$F' = (1-\gamma)F + \gamma E \tag{6}$$

where $\gamma \in [0,1]$ is the learning rate. Where $\gamma = 0$, no learning takes place and at the other extreme, $\gamma = 1$, the updated stochastic belief is completely determined by the last piece of evidence seen.

This abstract learning function requires further detail in both the PDF and credal set cases. This is developed below.

PDF updating

In the PDF case, the evidence must be transformed into a PDF as well. The evidence will have been the observation of a single event, say x_e. In the discrete event case, this evidence PDF can be encoded as:

$$e(x) = \begin{cases} 0 : x \neq x_e \\ 1 : x = x_e \end{cases} \tag{7}$$

Then the updated PDF that is used by the agent for future decision making is given by:

$$f'(x) = (1-\gamma) f(x) + \gamma e(x) \tag{8}$$

It is worth noting that the constraint $\sum f(x) = 1$ remains satisfied after applying this learning function.

Credal set updating

In a similar manner to the PDF approach, in this case the observed evidence must be transformed into a CDF. Again, if the observed evidence is given by x_e, the evidence CDF can be encoded by:

$$E(x) = \begin{cases} 0 : x < x_e \\ 1 : x \geq x_e \end{cases} \tag{9}$$

This produces a step-function, with the step rising at the evidence point. Note that this is simply the integral of the PDF version (Equation 7).

In the credal set case, there are now two CDFs that need to be considered, the lower and upper bounds of the credal set. In a similar approach, the learning algorithm is simply applied to both boundaries. The rationale for this is that if the same piece of evidence (or observation) was presented at each learning cycle, the credal set must converge to this evidence function.

$$\overline{F}'(x) = (1-\gamma)\overline{F}(x) + \gamma E(x) \tag{10}$$

$$\underline{F}'(x) = (1-\gamma)\underline{F}(x) + \gamma E(x) \tag{11}$$

Note that the updated functions are a linear weighted combination of two other CDFs and therefore they are also CDFs. Hence, these properly define a credal set.

Agent Implementation

Agents are used to model a set of individuals observing and acting within an environment. Ultimately, the aim is to test how well they are able to identify, through evidence based learning, suitable other agents for collaborating on a given task. For the purposes of this paper, the aim will be the slightly simpler task of learning to identify which other agents within the system an agent should prefer to collaborate with through forming a direct network link with.

Each agent forms a view (or belief) of the characteristics of all the other agents within the system. This belief is updated with evidence as and when it is observed by the agent. The agent uses the belief of the other agents' characteristic to determine which is most appropriate to interact with. If there are N agents within the MAS, then the data structure for each agent i is given by $(\mathscr{B}_{i1}, \mathscr{B}_{i2}, \ldots, \mathscr{B}_{iN})$, where $\mathscr{B}_{ij}$ is agent i's belief of agent j's characteristic distribution. The belief component can

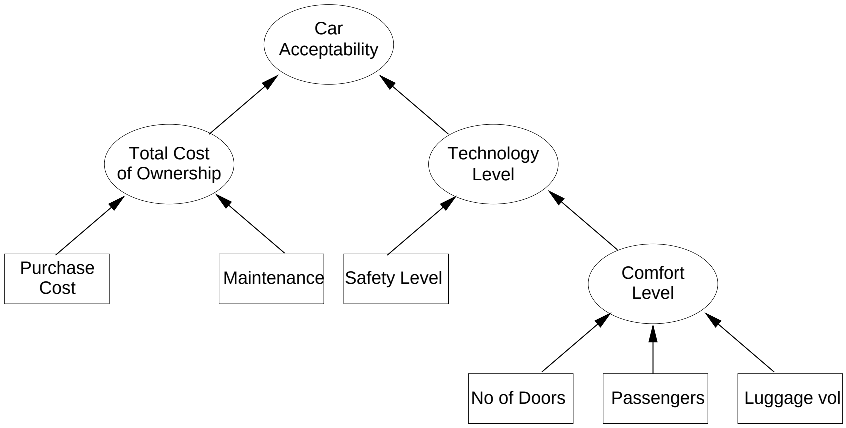

Fig. 2 Rule structure for the conceptual car domain [10].

be either a PDF or a credal set, and these will be compared in the empirical section of this paper.

To compare the PDF and credal set belief models, the UCI Car design database is used as a design domain model [10]. Here, each design variable is determined by an agent, and the agents need to identify suitable agents to interact with in order to complete a car design.

UCI Car Design Database

The UCI Car design database [10] provides a simplified representation of an automotive design domain. This domain consists of ten variables with a known hierarchical rule structure (see Figure 2). The design variables can be categorised as design parameters (controlled by the designer, shown in boxes) or design characteristics (function of the design variables, shown in ovals). As a corollary, the design parameters are the nodes of the rule structure. The database contains all the 1728 possible (legal) designs within this domain.

In this work, each variable was ‘assigned’ to a unique agent. The agents were not provided with any prior information about the rule structure, and therefore prior to any learning were completely unbiased to which agents they would prefer to collaborate with. Each agent was given a utility function that mapped variable state with ‘cost’ of moving to that state. These utility functions were not part of the original database, but needed as part of the MAS bidding system.

The database was used to generate design tasks. This was accomplished by randomly selecting a design from the database, thereby ensuring a legitimate design, and then transforming it into a design task. The design task was created by randomly blanking out a preset number of the design variables. The remaining set variables (four were left set for these simulations) represented the ‘design task’. The aim for the MAS was to then complete this design, i.e. define the blanked out design variables. It is possible that for any given design task, there were multiple possible complete satisfying designs, and so the aim was not to recreate the original randomly sampled design.

The design process was undertaken by a set of agents. Each agent had ‘control’ over a single design variable. The agents were able to observe each other through a blackboard approach [11]. The design task provided a global goal for the MAS. Within this, local goals were set for the individual agents in the form of the target variable setting for each agent. Individual agents must collaborate to be able to set or modify their own variable. Initially, the agents are given no information on the design variable structure and therefore must learn this through action and observation. When a pair of agents attempt to collaborate (i.e. change their variable settings), the overall success of this collaboration is used to update the belief of success for future collaborations.

The success of an agent-agent collaboration was measured using an abstract quality metric. For any collaboration, there were four possible outcomes, ranging from most successful to least: (1) the collaboration is a total success – both agents are able to move to the new variable setting; (2) the initiating agent is successful, the supporting agent is not; (3) the supporting agent is successful, the initiating agent is not; and (4) neither agent is successful in changing state. It is this collaboration outcome that is used to update an agent’s belief in its own suitability to collaborate with the other agent.

To reflect the logistical challenges that are inherent to physical design, each agent was augmented with a cost function. These cost functions were individually tailored for each agent and effectively represented the difficulty for an agent to achieve a specified variable state.

Each individual design task can be measured against a set of metrics. These metrics were used specifically to measure how well the MAS was learning as a result of the interactions involved in each design task. The metrics were:

Cost each design task incurred a cost as a result of the sum of the ‘interactions’ that occurred between the agents during the design;

Score each agent interaction was scored based on the degree of change in the design, ranging from 0 (no change) to 1 (both agents successfully changed state) and provides a measure of ‘quality of collaboration’;

Task completion was the number of design variables set. As the MAS was not able to backtrack the design process (for example by clearing previously set design variables) it could therefore find itself in a ‘dead end’, hence this provided a measure of how successful the MAS was in completing designs; and

Number of agent interactions any individual interaction was not necessarily going to be successful in modifying the design, and therefore the total number of interactions for a given design task measured how efficient any given design process was.

In addition to these basic metrics, two further metrics were derived:

Mean cost per variable set this is the average cost incurred in determining each design variable for a given design and

Mean interactions per variable set this is the average number of interactions required to set each design variable.

The mean cost metric is important due to the MAS not being able to backtrack. It is therefore possible that design cost reduces as a result of task completion reducing. By measuring the mean cost per variable set, this is able to illustrate if this is indeed the case. Note that it is not necessary to measure mean cost per interaction, as this is determined by utility function, and therefore will be independent of any learning and constant throughout the simulation.

The mean number of interactions per variable provides another measure to how efficient (in terms of ‘effort’ due to collaboration) the MAS is at any point. This basically measure how often an agent on average attempts to set a design variable before succeeding.

Empirical trials

The empirical trials seek to compare the MAS performance characteristics using a PDF belief model against the credal set belief model. This comparison will be undertaken using the UCI Car design domain. Due to the stochastic nature of these experiments, each individual trial must be repeated several times to get meaningful results.

The UCI Car domain has added complexity to reflect the complexities that are involved in real product design. The goal is to identify other agents that are suitable collaborators based on observed past performance. These observations arise as a result of attempted collaborations.

For each experiment, there is a small set of parameters. Principally, these parameters set the learning rate (γ) and the duration or the complexity of the task (N). For the Car design domain this is determined by the number of variables that were specified prior to the design task, and therefore the larger N, the fewer free design variable are left to be determined by the MAS.

The hypothesis that will be tested is:

H_0 There is either no significant difference between using the credal set belief model over the PDF belief model;

H_1 The credal set belief model and the PDF belief model are significantly different.

The car design experiments used a series of tasks. Each task specified the same number of design variables, but the exact variables and target values were randomly sampled. Each design task requested that the MAS identify a design solution with certain design variables set to given values. Each design task was generated by randomly selecting a design from the car database and then randomly selecting a subset of design variables.

Each agent has a belief of how well it is able to collaborate with each and every other agent. As there are 10 agents in the car design domain, each agent has 9 belief models. An agent will use their belief models to determine which other agent is the most likely to result in a successful collaboration. A collaboration is initiated by one agent who wishes to achieve a given result and a collaborating agent who will offer to help the initiating agent in obtaining that result. After an initiating agent has selected a collaborator and attempted to collaborate with that agent, it is able to see the result of this collaboration. There are four possible outcomes of the collaboration (in descending order of overall quality of outcome): (1) both agents successfully change; (2) the initiating agent is able to change, but not the collaborator; (3) the collaborator is able to change, but not the initiator; and (4) neither agent successfully changes. This outcome is mapped onto a numerical 'quality of collaboration' score which is in turn used as evidence for the initiating agent to update their belief model.

The key experimental variable was the belief model learning rate, γ. The key experimental outcomes that were measured were: (1) the average cost of setting each variable, (2) the average number of interactions per variable set, and (3) the average task completion level. In terms of 'optimal' outcomes, lower is better for cost and number of interactions while higher is better for task completion.

For this experiment, each independent design trial was run for 120 iterations. This run allowed the agents to refine their belief models. Each trial was repeated 30 times, and the average value is reported.

Results

Figures 3 to 9 represent how the car design MAS performs against the three metrics as γ ranges from 0.0 to 1.0 in increments of 0.1. These graphs clearly suggest that for most values of γ (the learning rate), there is little difference in performance of the car design MAS. At $\gamma = 0$, both systems undertake no learning and therefore this effectively represents the performance of the MAS at its initial state. From the graphs it can be seen that its performs poorly in terms of cost and number of interactions. Oddly, this case has the best (highest) task completion of all. For the remaining values of γ, there is little trend to be seen, and the systems appear to be stable for all γ values.

Figure 3 compares the convergence of the average cost per variable set of the PDF and credal set belief models. Initially, the credal set-based belief model performs worse than the PDF-based belief model, but by iteration 70 both approaches have converged in terms of this metric. Further, they both continue to perform at this level for the remainder of the learning process, suggesting that both systems have stabilised. This learning run was performed at the MAS datum parameter settings.

In a similar vein, Figure 4 compares the average number of interaction each agent performs per variable set. Under this metric, both the PDF and credal set-based systems initially perform at the same level. From iteration 50, a divergence is seen with the credal set-based system performing worse, however this appears to be temporary.

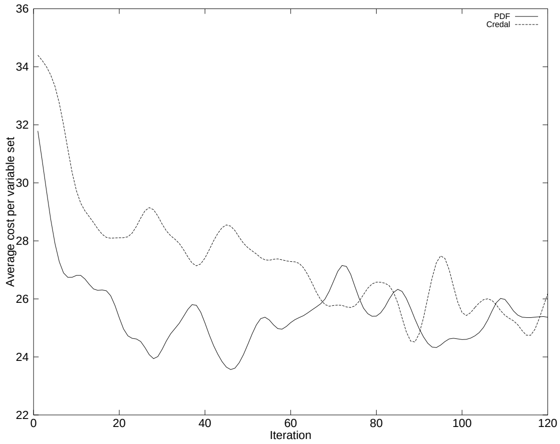

Fig. 3 Comparing convergence between the PDF and credal set representations of the average cost per variable set as learning progresses ($\gamma = 0.2$).

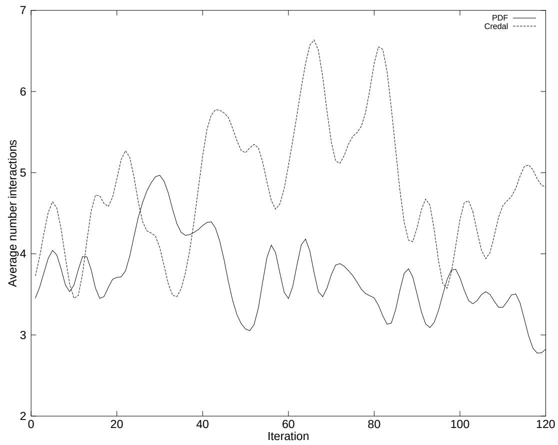

Fig. 4 Comparing convergence between the PDF and credal set representations of the average number of interactions per variable set as learning progresses ($\gamma = 0.2$).

It is also worth noting that the difference between these two systems is small: at the greatest divergence (ca iteration 80), the difference between them is 3 interactions.

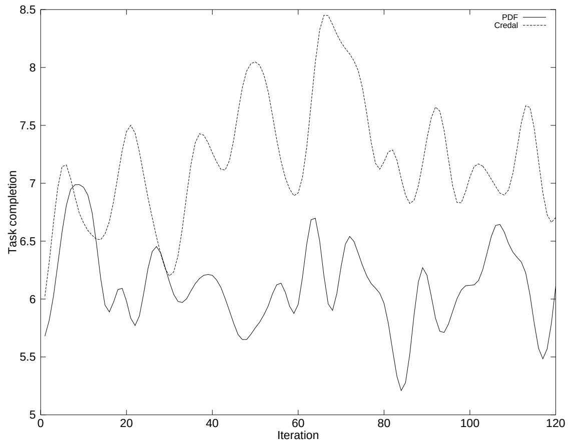

Fig. 5 Task completion as learning progresses ($\gamma = 0.2$).

A similar result is seen in Figure 5, the level that the design tasks have been completed as learning progresses. Both systems perform similarly, with the credal set system completing about one more design variable than the PDF system throughout the learning process.

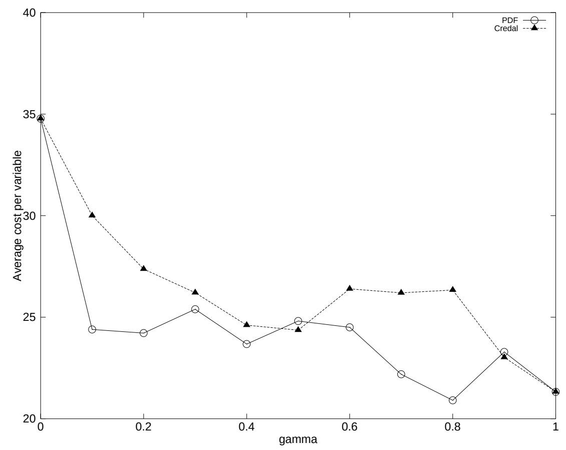

Fig. 6 Comparing average cost per variable set after learning for different γ values: PDF v Credal.

The following set of results compare the performance of the two systems as a function of the learning rate, γ. Within these results, it is worth noting that at the one extreme $\gamma = 0$, no learning occurs, whereas at the other extreme, $\gamma = 1$, the learning is completely based on the previous iteration's observation. In both these extremes, the two systems will perform identically.

The first of these results, Figure 6 compares the average cost of setting each design variable against the learning rate. On average, the credal based system performs slightly worse. However it is more consistent throughout the learning rate range, whereas the PDF based system rises again in the middle of the γ range.

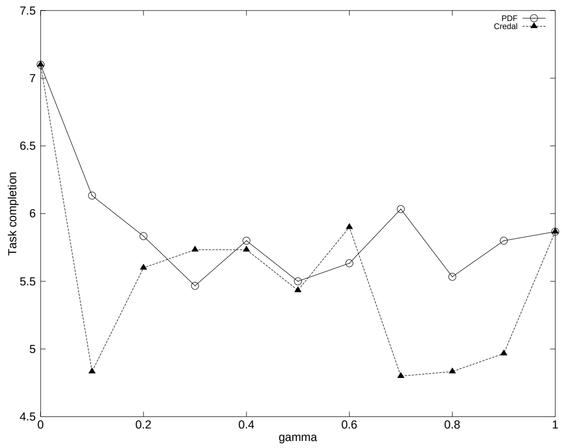

Fig. 7 Task completion against learning rate (γ).

Figure 7 illustrates the task completion level for both systems against the learning rate. Similar characteristics are again seen, with both systems converging for the middle of the γ range. Towards either extreme, the PDF based system does perform slightly better than the credal set system.

The final set of empirical trials were to compare how the two systems responded to different design task settings. These are independently illustrated for the PDF case (Figure 8) and the credal set case (Figure 9). Three values of N, the number of design variables that are predetermined (or how many variables are set as part of the design 'specification'), were tested. In both cases (PDF and credal set) it can be noted that the performance, in terms of average cost per variable set, decreases as there are fewer design variables to set. Again, in both cases, both systems converge to similar performance levels for the same task complexity.

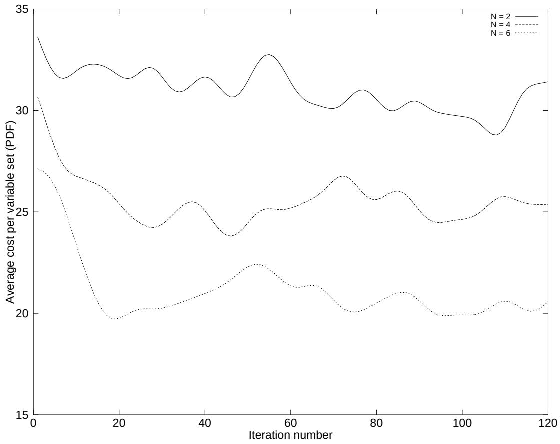

Fig. 8 Average cost per interaction as learning progresses for different design task complexities given by N, the number of variables that have been predetermined (PDF case, $\gamma = 0.2$).

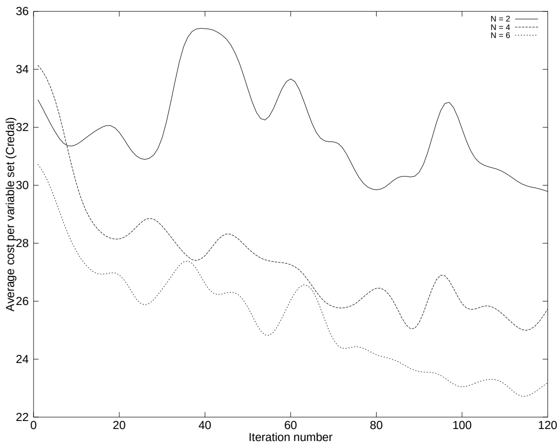

Fig. 9 Average cost per interaction as learning progresses for different design task complexities given by N, the number of variables that have been predetermined (Credal set case, $\gamma = 0.2$).

Discussion

The empirical work compared how a PDF based belief model performed against a credal set based belief model. The key metrics that were used to base the comparison on were: (1) how well the design MAS could complete the design task, (2) the mean cost for setting each design variable, and (3) the mean number of interactions required to set each design variable.

The experiments were based on the assumption that there was no prior knowledge to initialise the belief model. Therefore, in both cases the belief models were initialised uniformly. In the PDF case, this meant a uniform distribution model across all outcomes. On the other hand, in the credal set case, the belief models were initialised to be the full interval for all outcomes. Both cases used a similar learning algorithm, where evidence was used to modify the belief models to increase the probability of the recently seen evidence.

The key conclusion that can be drawn from this comparison is that a stochastically based decision support algorithm can use either the PDF or credal set representation of belief. In terms of the hypothesis presented earlier, H_0 would be accepted. Specifically, it means that where a PDF representation has been used in the past, this is able to be replaced by a credal set representation. As credal sets are more readily elicited from domain experts than PDF [7], this represents an important advance in decision support systems instantiation.

Conclusion

The key question this paper set out to investigate was to compare the effectiveness of pointwise versus interval probabilities in a collaborative engineering design process. To tackle this question, an abstract design environment was used themed on a multi-agent car design process. The car design was divided into a small number of loosely coupled sub-design tasks, and each of these tasks was undertaken by an independent agent. These agents needed to collaborate to be able to successfully complete the design.

The design agents were initially given no information as to which other agents within the system they could collaborate with. Through interaction with each other, the agents could gather evidence and learn which other agents they most successfully were able to collaborate with. Specifically, the evidence was used to update the agents' belief models of each others' capabilities.

The belief models were implemented using the two stochastic representations: pointwise and interval (credal set) probabilities. The computational experiments demonstrated that the credal set approach performed no worse that the (classic) pointwise representation. Given the benefits to be gained from using a credal set approach, such as ease of elicitation from domain experts, it can therefore be argued that there is a call to replace the use of pointwise stochastic models with credal sets in engineering contexts.

Further work is required to determine how sensitive the credal belief model is to the interval size given for each outcome. Clearly, as the interval of the credal set narrows, the credal set in the limit converges into the PDF. Therefore, it is important to know what the lower interval limit for an effective credal sets is. This will support the belief elicitation process in providing guidance as to how much information is required.

Acknowledgements

This research was undertaken while the author was visiting the University of Amsterdam Informatics Institute. Thanks must be given to the Royal Academy of Engineering and Thales Research Netherlands for supporting this visit. The conceptual basis for this work was developed while the corresponding author was a Fellow at the Institute of Advanced Study, Durham University.

References

1. Anja M. Maier, Matthias Kreimeyer, Clemens Hepperle, Claudia M. Eckert, Udo Lindemann, and P. John Clarkson. Exploration of correlations between factors influencing communication in complex product development. *Concurrent Engineering-Research and Applications*, 16(1):37–59, MAR 2008.
2. A. M. Maier, M. Kreimeyer, U. Lindemann, and P. J. Clarkson. Reflecting communication: a key factor for successful collaboration between embodiment design and simulation. *Journal of Engineering Design*, 20(3):265–287, 2009.
3. M J Chalupnik, D C Wynn, and P J Clarkson. Approaches to mitigate the impact of uncertainty in development processes. In M Norell Bergendahl, M Grimheden, L Leifer, P Skogstad, and U Lindemann, editors, *Proceedings of the 17th International Conference on Engineering Design*, volume 1, pages 459–470, Stanford, 2009.
4. N D Armoutis, P G Maropoulos, P C Matthews, and C D W Lomas. Establishing agile supply networks through competence profiling. *International Journal of Computer Integrated Manufacturing*, 21(2):166–173, 2008.
5. F Tonon. Some properties of a random set approximation to upper and lower distribution functions. *International Journal of Approximate Reasoning*, 48(1):174–184, 2008.
6. J M Aughenbaugh. The value of using imprecise probabilities in engineering design. *Journal of Mechanical Design*, 128(4):969–979, JUL 2006.
7. Peijun Guo and Hideo Tanaka. Decision making with interval probabilities. *European Journal of Operational Research*, 203(2):444–454, JUN 1 2010.
8. S Ferson and JG Hajagos. Arithmetic with uncertain numbers: rigorous and (often) best possible answers. *Reliability Engineering & System Safety*, 85(1-3):135–152, JUL-SEP 2004.
9. I Levi. *The enterprise of knowledge: An essay on knowledge, credal probability, and chance*. MIT Press, Cambridge, MA, 1980.
10. A Asuncion and D J Newman. UCI machine learning repository, 2007. University of California, Irvine, School of Information and Computer Sciences.
11. G W Tan, C C Hayes, and M Shaw. An intelligent-agent framework for concurrent product design and planning. *IEEE Transactions on Engineering Management*, 43(3):297–306, AUG 1996.