BAYESIAN NETWORKS FOR DESIGN

A stochastic design search method

PETER MATTHEWS
Durham University, UK

Abstract. A method for flexibly searching the conceptual design space using a stochastic approach is presented. From a database of previous design exemplars, a novel and inexpensive algorithm is used to induce a Bayesian Belief Network (BBN) that represents the causal relationships between a design domain's variables. This BBN is then used as part of an interactive tool for stochastically searching the conceptual design space using two search heuristics. This method is illustrated using a number of design scenarios based on a conceptual car design domain. The paper concludes with future research avenues to further the functionality of the BBN-based design search tool.

1. Introduction

The conceptual design stage occurs during the earliest parts of the design process. This is where a design specification is transformed into an abstract solution, representing the core concepts of the final design. The fluid nature of the conceptual design stage provides a challenge when developing deterministic models of a design at this phase. Specifically, it is difficult to explicitly define metrics for concept quality and this is left to the subjective expertise of the design team. The nature of conceptual design means that it is possible for a 'good' concept to be poorly detailed and thus result in a poor final product and *vice versa*. However, in general good concepts are more readily transformed into good final products while poor concepts require greater effort to attain a similar final high quality level.

A potential approach to this challenge is to adopt a stochastic perspective of the conceptual design phase. This allows for a more flexible representation of the design domain where multiple outcomes are possible. By using Bayesian Belief Networks (BBNs) to model a design domain, it is possible to work with partially defined design concepts. As more of the design is specified, the more accurate the model becomes at predicting how the remainder of the design is likely to be. An interesting and powerful

J.S. Gero (ed.), Design Computing and Cognition '06, 223–241.

aspect of the BBN is that it does not distinguish between the design parameters that are directly controlled by the designer and design characteristics which are determined as a result of the designer's decisions on the design parameters. This allows a designer to specify the characteristics at the outset and to then be guided towards design parameters that are likely to secure these characteristics.

This research has developed a method for inducing a BBN from a database of prior design exemplars using a novel information metric (Section 4). Once the BBN has been instantiated, a set of search heuristics are proposed to help guide a designer using the BBN to complete a partial conceptual design (Section 5). This method is illustrated using a set of design scenarios (Section 6). The paper concludes with a discussion of this method and some future development avenues for this stochastic approach.

2. Background

The first task in the design process can be argued as determining the specification of the final constructed artefact or product. The specification will be a combination of 'demands' that the design must fulfil and weighted wishes, which represent desirable but not essential aspects of the design. This specification can be expressed as a simple list of features (Pugh 1990) or encoded as an 'acceptability function' (Wallace et al. 1996). The specification guides the designer towards generating concepts that fulfil the demands. Alternative designs are discriminated between how well they either fulfil the wishes or evaluate against the acceptability function. Provided the specification does not impose overly restrictive demands, the designer is still left with a large conceptual design space to explore.

Conceptual design is by definition fluid. It is left to the detail and embodiment stages to crystalise the design into an artefact that can be manufactured (Pahl and Beitz 1996). A good concept will be easily transformed into a good final design. Conversely, a poor concept will require extensive effort to transform into a good final design. This definition of good/bad concept can only be measured after the final product has been produced, and is of little use during the conceptual stage of the design process. Also, the notion of a 'good' final design is domain and context sensitive. A designer will have a notion of what aspects of the final design are desirable, and a good designer will create concepts that are more likely to have these outcomes.

As a means for resolving the lack of explicit overall quality measure, an alternative, stochastic, view is adopted. This stochastic approach is fundamentally that a good concept has a high probability of resulting in a good final design, whereas a poor concept has a low probability of being transformed into a good design. This leads to a stochastic view of the design process: the probability of a good design at the end of the process depends on the quality of the initial design concept.

The fluidity of the conceptual design phase means it is difficult to provide concrete evaluation tools. Methods exist for creating ‘robust’ designs and, through objective evaluation techniques, guide the designer towards concepts that will be able to tolerate changes in the original specification (Taguchi et al. 1989; Ziv-Av and Reich 2005). In effect, these methods aim to provide the most generic design solution that is acceptable. These methods require a predefined evaluation function for the design that encodes the original design specification. An alternative stochastically driven approach is to bias towards design refinement that do not have ‘spiky’ probability distribution functions (PDFs). Such PDFs lack robustness as any deviation from the peak will result in a significant reduction in the likelihood of design success.

The approach taken in this paper is to provide guidance on the order that design variables should be determined. This designer guidance concept is similar to the Signposting methodology (Clarkson and Hamilton 2000), however it uses the shape of the dynamically computed PDFs rather than predefined domain rules to determine the order that the design variables should be determined.

An important aspect of this method is the inducing of domain models from previous design exemplars. The methods for creation of domain models can be represented on a spectrum ranging from expert based through to fully algorithmic. The expert based end of the spectrum provides high quality transparent models, however these require considerable time investment from domain experts which can be prohibitive. At the other extreme, pure machine learning methods tend to provide complex and opaque models, which while accurate, do not necessarily provide a designer with significant insight into the domain.

A motivating factor for this research is the cognitive aspects that affect human designers. These include the range of model complexities that can be intuitively handled; the nature of understanding a design domain; the latent differences between novice and expert designers; and what constitutes an intuitive interface to a stochastically based design domain model.

3. Bayesian Design

Bayesian design is the use of Bayesian Belief Networks to support the design process. Bayesian Belief Networks (BBNs) provide a causal model for a set of observations or variables (Jensen 2001). These models are represented graphically, where the observations are the graph nodes and the causal links are the directed edges that connect the nodes. As the networks tend to be relatively sparse, namely that nodes are typically only attached to a small subset of other nodes, this significantly simplifies the computational effort required to make inferences given a set of observations. As observations are made, these provide information for the model. The model uses these observations to make informed estimates on the values of the non-observed variables. For a non-observed variable, it is possible to compute its informed (conditional) probability distribution function. Effectively, the available information biases the unobserved variable's PDF.

In the design context, the observed variables are the design parameters and characteristics. The distinction between these is primarily that design parameters are directly determined by the designer while design characteristics are a result of the design parameters. For the purposes of this work, no distinction is made between these two, as it is impossible in general to infer the causal order between the design variables. For example, when designing a bridge one of the design parameters is the width of the bridge. The wider the bridge, the greater the potential flow across the bridge which is a design characteristic of the bridge. However, a greater potential flow across the bridge will require a stronger bridge, which can be achieved through a number of alternatives, e.g. material choice, structural design, etc., all of which are design parameters again.

Bayesian design is a stochastic view of design, and is particularly appropriate for routine early design tasks. Due to the fluid nature of the early design phases, this is an appropriate approach. Under the stochastic view, each design variable has a PDF. This PDF is a mapping from the values the design variable can take (design space) to the probability of that variable taking that value. The probability of a variable taking on a particular value represents is a measure of how frequently that variable takes that value in final (e.g. detail phase) designs. This can be interpreted as a measure of the design knowledge or experience that exists for achieving the given design variable value. Thus, where low probabilities are encountered, this provides a warning that a potential challenge lies ahead in achieving that position in the design space.

As these PDFs are computed within a BBN, these will be biased where relevant information is available. Relevant information in this context are observations taken from neighbouring nodes within the network. The updated conditional PDFs (CPDFs) now take into account the knowledge that exists about a subset of designs from the domain, as defined by the relevant information that has been added. So where previously setting a design variable to particular value might have appeared difficult to achieve by nature of the low probability of this outcome, it is possible that given the additional information this is becomes a much more likely outcome.

This leads into exploiting design BBN as a design support tool. A designer will start with a specification that defines a subset of the design variables. These defined variables can be considered as observations and thus be entered into the BBN. The BBN can now provide CPDFs for the unobserved variables. These unobserved variables were not part of the specification, and hence it may be assumed that the designer is free to set these arbitrarily. The designer wishes to produce a design concept that will have the greatest chance of producing a good concept, as these are least likely to require extensive effort during the detailing phases to produce a good final design. Hence, the designer should be attracted to set design variables to their most likely states, as these represent the states where the most knowledge and/or experience exists.

Where a number of different variables require determining, a simple ordering heuristic can be applied. Design variables with narrow 'spiky' distributions should be determined first, proceeding through until the variables with the 'flattest' PDFs being last. This ensures that design variables with narrow likely ranges are set suitably as early as possible. If this is not done, it is possible that through the setting of another design variable, the 'narrow' design CPDF disappears altogether, thus representing a highly unlikely design. In effect, this is the stochastic equivalent of over constraining a design. Similarly, the 'flat' PDFs are likely to become spikier as more of the design is defined. By monitoring how each individual PDF changes with each additional design variable setting, it is possible to dynamically guide a designer through the order in which the design variables should be set. It is worth noting, however, that these are no more than guiding heuristics. Designers are at liberty to navigate through the design domain based on their personal experience or instincts.

4. Inducing Bayesian Networks

To use Bayesian Belief Networks as a design support tool, it is essential to acquire a good BBN in the first instance. The first step to achieve this is the creation of a suitable representation or encoding of the design domain. This provides a definition of the conceptual design space of the domain under consideration. A simple, but suitable, representation format is a design vector. The design parameters and characteristics form the variable components of the vector. As discussed in the previous section, these are to be the nodes of the BBN.

The next step is identifying the causal links between these design variable nodes. One method for achieving this is to use an expert (or panel of experts) to manually identify the links. While this is expected to produce accurate models, it is a time consuming exercise. As the domain becomes more complex in terms of number of design variables, the complexity of the model creation increases quadratically with the number of design variables. Further, once the nodes have been linked, the PDFs and CPDFs that are associated with the nodes and arcs respectively must be defined. This requires considerably greater consideration than identifying the causal links. As a result, the expert crafted BBN is not appealing.

An alternative method for identifying the causal links in the BBN is to apply data mining techniques to a database of previous design exemplars. These techniques analyse the given database and create a network that provides a sufficiently close representation of the stochastic phenomenon observed in the database. These algorithms use three main metrics to determine accuracy: *validity*, *understandability*, and *interestingness* (Mitra and Pal 2002). Validity measures what proportion of the data can be covered by the model. Understandability provides a complexity measure that can represent how easy it is for a designer to understand a model. Finally, interestingness measures the novelty of representation of a model in a design domain. These metrics have been listed in order of difficulty of measuring. Validity can be measured directly against the database supplied. Understandability requires a measure of human ability to understand a given model. Interestingness must be measured against the current state of domain knowledge and combined with a subjective element supplied by the domain expert.

4.1. INFORMATION CONTENT BASED METRIC

Most efficient BBN inducing algorithms require that the overall causal order is known prior to running the algorithm. However, where this ordering is not known, the complexity of most BBN graph inducing algorithm explodes to $O(n!)$, where n is the number of variables. In this research, it is assumed that the causal order of the design variables is not known prior to running the algorithm. A novel greedy algorithm has been developed for this work that reduces the computational complexity down to $O(n^2)$. This breadth-first greedy approach has been tested on some well known databases and performs well in terms of identifying the correct BBN. The overall process is illustrated in Figure 1.

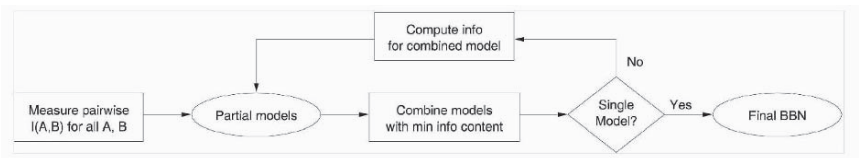

Figure 1. Flowchart representing the greedy BBN learning algorithm.

The graph search algorithm implements a greedy search heuristic based on a measure of the information content of the conditional probability distribution. Recall the definition of conditional probability:

$$P(B = b \mid A = a) = \frac{P(B = b, A = a)}{P(A = a)} \quad (1)$$

Where the events A and B are independent, $P(B, A) = P(B)P(A)$. Hence, when A and B are independent $P(B|A) = P(B)$. By considering the difference between the *observed* conditional and prior probability distributions, it is possible to measure the mean variation in this difference:

$$I(A,B) = E[P(B \mid A) - P(B)]^2 \quad (2)$$

The variation, I, represents how much more information is contained in the conditional probability distribution above the information contained in the prior probability distribution. A large value for I indicates that the conditional probability distribution contributes greatly to the knowledge of the domain while a small value indicates that the two variables are likely to be relatively independent of each other.

The graphical model search algorithm begins by measuring the pairwise information content between each variable pair. This is computed for both directions as in general $I(A, B) \neq I(B, A)$. For each design variable, the system is seeded with a partial model containing the given variable and the variable that has the greatest information content of its conditional probability distribution. Where a partial model would be repeated, the variable with the next highest information content is selected.

These partial models are ordered in increasing information content order. The next step is to merge partial models with low information content, creating a new partial model whose information content is given by the sum of its parts. The two lowest information content scoring models with a common variable are merged, resulting in one fewer partial models. Where there are more than two candidate models for combining, the tie breaker is determined by (1) resulting model complexity followed by (2) lower information score. This is repeated until all partial models are exhausted. The above greedy algorithm results in a single graphical model.

5. Implementation

To test the above design heuristics, it was necessary to implement the stochastic algorithm. To ensure wide access to the algorithm, it was decided to implement the interactive design support tool using Microsoft's Visual Basic (VB) within Excel. Most office desktops have access to Excel, and thus a large population of potential beta-testers exists.

The code is structured in two parts: The first part is a one-shot machine learning algorithm that uses Equation 2 to induce the network from a given dataset of prior design exemplars. As this only needs to be run once, it was written in Matlab rather than VB. While this restricts the ability for arbitrary users to use their own dataset, this is not a part of the user trial. The second part of the code represents the user interface to the BBN. Figure 2 contains the flowchart for the iterative and designer led search process. This is encoded as a VB macro that reads the current design state from the Excel design spreadsheet and computes the PDFs of the unspecified design variables. These PDFs are extracted from the database of design exemplars that resides on a separate spreadsheet. The conditional PDFs are computed from the joint probabilities that can be extracted by frequency counting within the database. The remainder of this section will focus on the user interface.

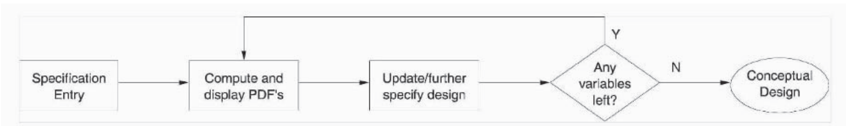

Figure 2. Flowchart representing the overall design search process.

5.1. DATA STRUCTURE

The data structures for the interactive design search tool are based on the simple native structures available within Excel. There are three types of data that need storing: (1) the database of previous design exemplars; (2) the network structure; and (3) the current design state. Each of these is held in a separate Excel worksheet. While this is not a highly efficient approach, it does provide a very simple representation that can be easily manipulated by a designer. Typically, a designer would be only interested in the design status worksheet. However, the designer also has the capacity to edit the BBN directly in the case that it is believed to be inaccurate. Also, the designer is able to edit the exemplar database, either by removing data points or adding further ones. However, if the manually edited data had an impact on the network, this would not be possible for the user to determine directly.

The design status worksheet lists each design variable on a separate row. The first column contains the variable name. In the next column, the variable value is placed, when known. The remaining columns are used to display the PDF for the given variable. The PDF is computed for all possible values the design variable can take. This is a simple task, as the all the design variables have been discretised and so there are only a small number of values to consider. The designer then uses the PDFs as a guide to determining the next design variable value.

Similarly to the design status worksheet, each row of the network worksheet contains the network data for a single variable. The first column contains the variable name. The remaining columns contain the immediate causal ‘parents’ of the variable. For each variable, X, these represent the set of variables that X is causally dependent on. This set of parent variables is typically denoted $\pi(X)$. Hence, in the BBN, the CPDF of X is expressed by $P(X \mid \pi(X))$.

Finally, the dataset work sheet simply contains a set of previous exemplar designs. Each design is listed on a separate row. The columns in this case contain the different design variables.

5.2. INTERACTIVE ALGORITHM

The interaction between designer and the code is centered around the unspecified design variables. For illustration purposes, denote the unspecified design variable as Y. To provide direct guidance, the information supplied for each unspecified design variable is reduced to a single dimension, namely the PDF for that design variable. Depending on the status of adjacent design variables, there are two main cases to be considered: (1) Y is a non-terminal node in the BBN tree and (2) Y is a terminal node. The BBNs that are induced from the greedy learning algorithm are tree structures: no node has more than one child, or alternatively, any variable can causaly only affect one other variable. However, a variable can have several parent variables that have a causal effect on it.

The first case is straightforward. The aim here is to compute the CPDF defined by $P(Y = y \mid \pi(Y))$ for all y values that the design variable Y takes. The CPDF only uses the specified parent design variables. That is, if one of the members of $\pi(Y)$ has not been specified, it is excluded from consideration. Clearly, if none of the parents have been specified, then the CPDF reduces to the PDF of the design variable Y.

In the second case, where the unspecified design variable Y is a terminal node, the code considers the child node of Y. As the BBN is a tree graph, there is only one child of Y. Let $X = \pi{-}1(Y)$ be the unique child of Y. The designer is then presented with the following distribution:

$$P(X \mid Y = y, \pi(X)) \tag{3}$$

There are now two further sub-cases to consider: X has been specified and X has not been specified. Where X is known, the algorithm proceeds to compute the probabilities of achieving this specified value for all possible values $Y = y$ that the unspecified design variable can take. Again, only the known values of $\pi(X)$ are considered. In the second case, where X has not been specified, the only information that can be used to guide the designer is the PDF of the unspecified variable Y. This is as Y is a terminal variable, so there are no further parents that will affect it, and it is independent to the other parents of X, namely $\pi(X)$.

It should be noted that in this second case, Equation 3 is not a proper PDF as it does not necessarily sum to 1. This function measures the likelihood of achieving the already determined value of X. However, for the purposes of identifying a good value for Y, the same argument applies, namely that a designer should focus on those values that provide a suitably high probability for achieving X's value.

All the PDFs are computed dynamically at run time by counting suitable exemplars from the database. The complexity of this process is $O(Nn)$, where N is the size of the database and n is the dimensionality of the design space.

5.3. DESIGNER HEURISTICS

The final aspect to be considered is how the displayed PDFs are interpreted by the designer as heuristics for the design search process. For each unspecified design variable, the relevant PDF for that variable is displayed in the columns adjacent to the design specification. As argued earlier, it is suggested that the designer focuses first on the variables with narrow distributions and then moves onto variables with ever wider distributions. This is the variable ordering heuristic. The second heuristic guides the designer to the value that each variable should be set to. It is suggested that the designer selects the value that has an acceptably high probability associated with it. This represents the most likely outcome for the design, or conversely, the design with the greatest likelihood of success.

6. Case Study: Preliminary Car Design

As an initial trial of the stochastic design search method, the well known UCI machine learning car design database was used (Blake and Merz 1998). This database contains a sample of 1728 designs, each with a full set of observations. Each sample represents a conceptual car design. The cars are represented as a 10-dimensional vector comprising of both design parameters and design characteristics. The design parameters are: the target purchase price; the expected maintenance cost; the designed safety level; the number of doors; the number of passengers; and the volume of luggage that can be carried. The design characteristics are: the overall cost of ownership; the comfort level; the technology level; and the overall car acceptability. All the design variables are discrete. A set of predetermined rules was used to map the design parameters onto the design characteristics to create the database that was then used by the greedy BBN induction algorithm. The structure of these rules is given in Figure 3. These structured rules provide a means for comparing the stochastic design tool to the original and defining structure of the design space.

The car database was first loaded into Matlab and passed to the BBN learning algorithm. This generated a network representing the causal links between the design variables. The algorithm produces exactly as many arcs as there are design variables. This resulted in a non-tree structure. In a tree structure each node, with the exception of the root node, should have a single child. The structure that was produced by the learning algorithm had the ‘safety’ node linked to both the ‘technology’ and ‘car acceptability’ nodes. By considering the information content of the two arcs coming out of the safety node, the arc with the lower information content was deleted. The resulting tree network that was learnt from the dataset had an identical causal structure to the underlying rule structure used to create original the design database, as illustrated in Figure 3. This network was then encoded in the Excel spreadsheet, along with the database.

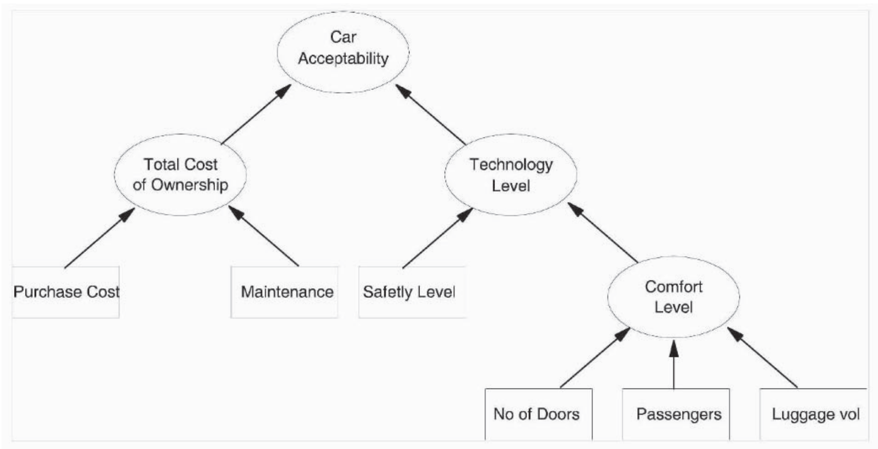

Figure 3. Rule structure for the conceptual car domain.

6.1. STOCHASTIC SEARCH

The Excel spreadsheet provides the ‘user interface’ to the stochastic design tool. Using this tool, four different design scenarios were explored based on the nature of the design specification: (1) only design parameters specified; (2) only design characteristics specified; (3) both specified; and (4) an ‘infeasible’ design specified. These are expanded below.

6.1.1. Design Parameters: ‘People carrier’

In the first scenario only design parameters (design variables under direct control of the designer) were specified. Specifically, a subset of the design parameters were specified to reflect a partial set of the requirements of a ‘people carrier’ type car. The design specification required that the car should have low maintenance costs, a high safety rating, seat a large number of passengers, and have a large luggage space. This specification omitted the design parameters describing the purchase price of the car and the number of doors.

This specification was entered into the spreadsheet, and the VB macro computed the PDFs for the unspecified design variables. Figure 4 is a screen shot from this step. The stochastic design heuristic suggests considering the design variables with the smallest distribution first. Further, to maximise the likelihood of the design, the heuristic suggests selecting the values that maximise this PDF. In this case, the order and settings of the design variables were guided as follows (see also Table 1):

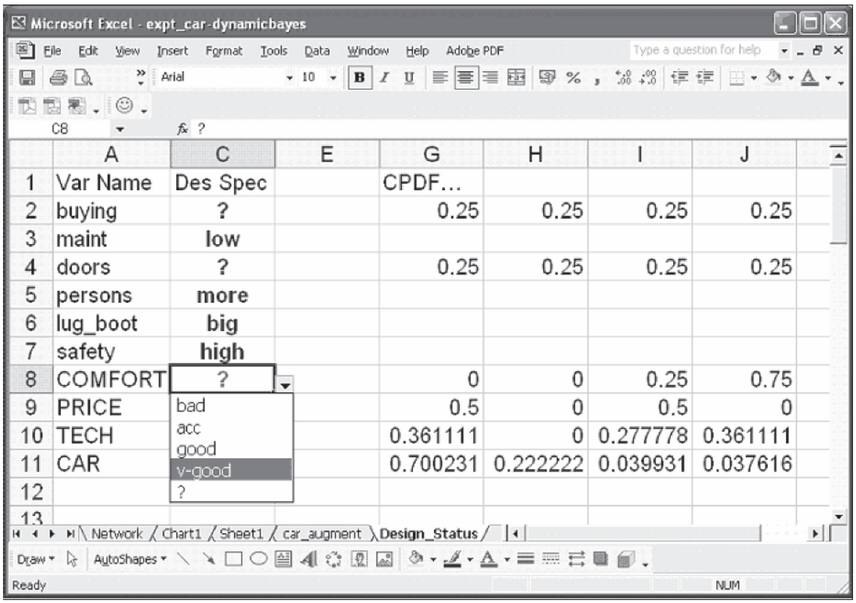

Figure 4. Screen shot from the 'People Carrier' design specification and initial PDF computation.

1. Comfort: set to 'high'
2. Technology: set to 'high'
3. Price: set to 'low'
4. Car acceptability: set to 'high'
5. Purchase price: set to 'low'
6. Doors: set to '5'

It must be noted that at each step there were other potential alternatives that could have been selected. Further, after each step, the PDFs for the remaining undefined variables did change, thus illustrating the dynamic nature of this search tool.

The final design does reflect a highly acceptable 'people carrier' design concept. This would then be taken through to a more detailed design phase.

6.1.2. Design Characteristics: 'Sports car'

The 'sports car' design scenario only specified the desired characteristics of the final design concept. Three design characteristics were specified: the car only required relatively low comfort, a high technology level, and a high overall ownership cost. This left a large number of design variables to be specified, which could potentially lead to infeasible designs without any guidance, Table 1.

TABLE 1. Search path for the unspecified design variables for the 'People Carrier'. Selected variable/value in bold.

Step	Variable	PDF/Likelihood			
1	Buying	0.5	0.25	0.25	0.25
	doors	0.25	0.25	0.25	0.25
	COMFORT	0	0	0.25	**0.75**
	PRICE	0.5	0	0.5	0
	TECH	0.6	0	0.28	0.36
	CAR	0.70	0.22	0.04	0.04
2	buying	0.25	0.25	0.25	0.25
	doors	0	1	1	1
	PRICE	**0.5**	0	0.5	0
	TECH	0	0	0	**1**
	CAR	0.70	0.22	0.04	0.04
3	Buying	1	1	0	0
	doors	0	1	1	1
	CAR	0	0	0	**1**
4	**buying**	**1**	1	0	0
	doors	0	1	1	1
5	**doors**	0	1	1	**1**

In a similar process to the previous scenario, the design heuristics suggested the following course of action, see also Table 2:

1. Safety: set to 'high'
2. Car acceptability: set to 'low'
3. Luggage space: set to 'low'
4. Purchase price: set to 'high'

In this scenario, there were two occasions where the shape of two PDFs were identical, Table 2, step 3, thus not providing a clear precedence for determining the values. In these cases, it is for the designer to use their discretion to the order of determining the values.

The final design, while appearing to score poorly on a number of characteristics, is in line with a high performance sports car that has traded off mass appeal against a niche market.

TABLE 2. Search path for the unspecified design variables for the 'Sports car'. Selected variable/value in bold.

Step	Variable	PDF/Likelihood			
1	buying	0	0	0.25	0.5
	maint	0	0	0.25	0.5
	doors	0.33	0.33	0.22	0.22
	persons	0	0.5	0.33	
	luggage	0.58	0.25	0	
	safety	0	0	**1**	
	CAR	1	0	0	0
2	buying	0	0	0.25	0.5
	maint	0	0	0.25	0.5
	doors	0.33	0.33	0.22	0.22
	persons	0	0.5	0.33	
	luggage	**0.58**	0.25	0	
	CAR	**1**	0	0	0
3	**buying**	0	0	0.25	**0.5**
	maint	0	0	0.25	0.5
	doors	0.33	0.67	0.67	0.67
	persons	0	1	0.75	
4	**maint**	0	0	0	**1**
	doors	0.33	0.67	0.67	0.67
	persons	0	1	0.75	
5	**doors**	**0.33**	0.67	0.67	0.67
	persons	0	1	0.75	
6	**persons**	0	**1**	0	

6.1.3. Design Parameters and Characteristics: 'Accessible luxury'

The 'accessible luxury' design scenario specified a combination of design parameters and characteristics. The specified design parameters were: the car should have low maintenance costs; be a four-door design; and have a high safety level. The car was to have the following characteristics: it should have a high comfort level and it should have a high acceptability level.

The stochastic search method suggested the following course of action, Table 3:

1. Technology level: set to ‘very high’
2. Luggage space: set to ‘high’
3. Overall cost of ownership: set to ‘low’
4. Passengers: set to ‘4’
5. Purchase price: set to ‘low’

In this scenario there were occasions where the guidance to selecting the variable value was ambiguous. For example, determining the overall cost of ownership placed equal weight between selecting ‘low’ or ‘high’ (see Step 3 in Table 3). In this case, as the car is intended to be ‘accessible’, the designer selects ‘low’. Had the designer selected ‘high’, this changes the options that are offered two steps later when selecting the purchase price where the designer is offered ‘high’ or ‘very high’.

TABLE 3. Search path for the unspecified design variables for the ‘Accessible luxury’. Selected variable/value in bold.

Step	Variable	PDF/Likelihood			
1	buying	0.25	0.25	0.25	0.25
	persons	0	0.33	0.67	
	luggage	0	0.33	0.67	0
	PRICE	0.5	0	0.5	**1**
	TECH	0	0	0	
2	persons	0	0.33	0.67	0
	luggage	0	0.33	**0.67**	
	PRICE	0.5	0	0.5	
3	buying	0.25	0.25	0.25	0.25
	persons	0	1	1	0
	PRICE	**0.5**	0	0.5	
4	buying	1	1	0	
	persons	0	**1**	1	0
5	**buying**	**1**	1	0	0

6.1.4. Infeasible Design

In the final scenario, an infeasible design was specified. The design was determined to be infeasible according to the rules that map the design parameters onto the design characteristics. Specifically, given sufficient design parameter information to determine the value of a characteristic, the characteristic was set to a different value thus representing an 'infeasible' design. While this is a slightly artificial case, it serves to demonstrate how the search method proceeds under such circumstances.

In this scenario, the stochastic search method reported a flat zero PDF for the cost of ownership characteristic (see Step 1 in Table 4). This indicates that under the current specification, there is no previous knowledge on what the likely cost of ownership for this design will be. If not having this information was acceptable, the designer could proceed with the current specification and find out further downstream in the design process what the value of this characteristic would be. As part of this case study, this option is not available. The alternative is to modify some other aspect of the design until a non-zero PDF arises.

To search for an alternative design specification that provides a non-zero PDF, the design domain BBN is used to track the parent and child variables of the cost of ownership variable. These are the purchase price, the maintenance cost, and the overall car acceptability. The design specification did not include either the purchase price or the maintenance cost, leaving the overall car acceptability design variable as the cause of the zero PDF. This leaves the designer with two options: either modify the child variable (i.e. the car acceptability) or consider the other parents of the child (in this case the technology level, Figure 3). This is as the PDF displayed for the overall car ownership variable is actually the CPDF of its child node, ranging over all possible values that car ownership can take. As such, the displayed PDF is the likelihood function, however the same value selection heuristics apply.

In this scenario, the designer decides to modify the comfort level variable. The designer slackens the specification on this variable until the PDF for ownership cost becomes non-zero, indicating that the design specification is feasible, Step 2, Table 4. Once the partial specification is feasible (no constant zero PDFs), the design search process continues as in the other (feasible) design scenarios.

6.2. NOTES ON TRADITIONAL SEARCH

A traditional approach to completing the design specification would in the first instance need to consider the design parameters and characteristics separately. While specifying the design parameters remains possible, as this is done directly by the designer, no information is made immediately available regarding the likely values the design characteristics would take on. These design characteristic values are only to be obtained if the designer has knowledge about the relationship between the design parameters and the characteristics. Without this knowledge, a designer must determine all design parameters and then obtain the design characteristics through more costly detail analysis or prototyping.

TABLE 4. Search path for the unspecified design variables for the 'Infeasible design'. The first step involves slackening the 'Comfort' variable. Selected variable/value in bold.

Step	Variable	PDF/Likelihood			
1	buying	0.25	0.25	0.25	0.25
	maint	0.25	0.25	0.25	0.25
	PRICE	0	0	0	0
	(COMFORT@v-high)				
2	buying	0.25	0.25	0.25	0.25
	maint	0.25	0.25	0.25	0.25
	PRICE	**1**	0	0	0
	(COMFORT@high)				
3	**buying** maint	0.5	**0.25**	0	0
		0.5	0.25	0	0
4	**maint**	**1**	0	0	0

The reverse approach where the designer specifies the design characteristics and then searches for appropriate design parameters is not directly possible with a traditional search. Where no or little knowledge exists, the designer must guess initial design parameter settings and then test. This must be repeated until either a sufficiently good design is achieved or enough knowledge is generated to be able to understand the design domain sufficiently well for the purposes of meeting the specification.

Both these approaches require performing extensive number of experiments where the designer lacks knowledge on the nature of the relationships between the various design variable.

7. Discussion

There are two aspects to this stochastic design search method: inducing the BBN design model from previous design exemplars and using the BBN as a search tool. The information based induction algorithm appears to perform well, based on a series of tests using databases taken from known source models. The car design database provided an example of this, where it identified the network structure with a single extra arc. This spurious arc was easy to identify, as it was the arc with less information from one of two potential arcs that broke the tree structure.

Using the BBN induced from the design database as a dynamic search tool offers an efficient search strategy when the two search heuristics are employed. The feasible design scenarios mainly followed the search heuristics, with the designer rarely ‘deviating’ from the first ranked choice. Further trials are needed where the designer does not follow these suggestions.

Where a designer starts with an infeasible design, as per the final design scenario, the stochastic search tool simply reports constant zero PDFs for the unspecified variables. In the reported scenario, the designer used knowledge of the BBN structure to identify the ‘neighbouring’ design variables to modify blindly. An improvement would be to provide some form of guidance to identify fruitful modifications to the current partial design specification. This would allow the designer to ‘unblock’ the infeasible design specification using a minimal change to the original specification.

8. Conclusions and Future Work

Using the Bayesian Belief Network with the two search heuristics provides an efficient conceptual design search tool. The two heuristics aid the designer to first identify the next design variable that should be determined, followed by which value would provide the most robust design. A powerful aspect of the BBN approach is that the designer need not distinguish design parameters from design characteristics. This allows a designer to specify design characteristics that are not normally under a designer’s direct control. However, it must be emphasised that the designer is not constrained by the design heuristics and is free to explore the design space in other orders. This offers the designer the flexibility that is essential during the conceptual design stage.

Further work is required in a number of areas. Research is needed on how to develop a more intuitive user interface to the BBN. There is a need for metrics for PDF ‘spikiness’ versus ‘flatness’. This is critical as it will not be possible for a designer to identify the narrowest of PDFs in a design domain with considerably more variables. Another key area for further work is to develop methods for identifying design variables in infeasible design specifications that could be fruitfully slackened. Currently, the designer only has the network to identify neighbouring variables but no information on which variable should be modified.

Finally, this work was based on an artificial database with a fully tested set of designs (in terms of the design parameters). Further investigations are required where this is not the case, as this represents real design situations.

Acknowledgements

This research is funded by a Nuffield Foundation Award to Newly Appointed Lecturers in Science, Engineering and Mathematics (Grant number: NAL/00846/G).

References

Blake, CL and Merz, CJ: 1998, UCI Repository of machine learning databases, Available Online: http://www.ics.uci.edu/˜mlearn/MLRepository.html.

Clarkson, PJ and Hamilton, JR: 2000, Signposting: a parameter-driven task-based model of the design process, *Research in Engineering Design* **12**(1): 18-38.

Jensen, FV: 2001, *Bayesian Networks and Decision Graphs, Statistics for Engineering and Information Science*, Springer, NY.

Mitra, S and Pal, SK: 2002, Data mining in soft computing framework: A survey, *IEEE Transactions on Neural Networks* **13**(1): 3-14.

Pahl, G and Beitz, W: 1996, *Engineering Design: A Systematic Approach*, Springer-Verlag London, 2nd ed.

Pugh, S: 1990, *Total Design*, Addison-Wesley.

Taguchi, G., Elsayed, EA and Hsiang, TC: 1989, *Quality Engineering in Production Systems*, McGraw-Hill.

Wallace, DR, Jakiela, MJ and Flowers, WC: 1996, Design search under probabilistic specification using genetic algorithms, *Computer Aided Design* **28**(5): 405-421.

Ziv-Av, A and Reich, Y: 2005, SOS — subjective objective system for generating optimal product concepts, *Design Studies* **26**(5): 509–533.