P. C. Matthews *

Advanced Mechanics Research Group, School of Engineering, Durham University, Durham DH1 3LE, UK

A Bayesian support tool for Morphological Design

Abstract

Dynamic computer based support tools for the conceptual design phase have provided a long-standing challenge to develop. This is largely due to the 'fluid' nature of the conceptual design phase. Design evaluation methods, which form the basis of most computer design support tools, provide poor support for multiple outcomes. This research proposes a stochastic-based support tool that addresses this problem. A Bayesian Belief Network (BBN) is used to represent the causal links between design variables. Included in this research is an efficient method for learning a design domain network from previous design data in the structure of a morphological design chart. This induction algorithm is based on information content. A user-interface is proposed to support dynamically searching the conceptual design space, based on a partial design specification. This support tool is empirically compared against a more traditional search process. While no compelling evidence is produced to support the stochastic based approach, an interesting broader design search behaviour emerges from observations of the use of the stochastic design support tool.

Key words: Bayesian belief network, data mining, decision support, conceptual design, information content

Preprint submitted to Elsevier Science *July 30, 2008*

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*: a 'poor' concept can be carefuly developed through the detailing phase to result in a 'good' final product. The terms 'good' and 'poor' in this case are context dependent, and cover such criteria as technical quality, commercial success, and aesthetics. 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.

For the purposes of this research, a working definition of the conceptual design stage is required. There is general agreement that the conceptual stage is where designers transform the initial technical specification into a general form that defines the overall aspects of the product [1,2,3]. A core aspect of the research presented here relies on using machine learning techniques that require databases of prior design examples. These databases of conceptual designs require defining, and hence there is a need to define conceptual design. In this context, conceptual design will be considered in a morphological sense: the design will be structured into a set of parametric and characteristic vari-

* Phone: +44 191 334 2538; fax: +44 191 334 2407
Email address: p.c.matthews@durham.ac.uk (P. C. Matthews).

ables [1,4,5]. Against each of these variables, a designer will have to select a conceptual solution. The combination of all these solutions then generates the final design concept. Generating 'good' design using this approach relies on the designer understanding the interaction between the design variables.

A potential approach to the challenge of understanding the relationships between the design variables is to adopt a stochastic perspective for 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 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.

The BBN guided design search process claims four key effects. First, that a satisfactory design domain model can be induced from previous design exemplars, and that this explicit design model is of use to a designer. Second, that it is possible to start the design search based on a partial design definition that may include design characteristic values as opposed to being required to initiate the search by estimating design parameter values. Third, the design search heuristics for using the BBN suggest an efficient path through the design process. Fourth, the resulting designs are produced efficiently and are of similar or higher quality than designs produced without this BBN support tool. These claims will be illustrated and tested within this paper.

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, an interactive search tool is used that dynamically guides the designer to the most likely design based on the current partial design state (Section 5). This search method is extensively illustrated using a design scenario that forms the central application of this paper (Section 6). The empirical work seeking evidence of the benefits of this method is then described (Section 7). The paper concludes with a discussion of this method and some future development avenues for this stochastic approach.

To illustrate the ideas being developed, three examples are provided. First, a conceptual bridge design is used as a means to describe the overarching concepts that form the basis of the Bayesian design support tool. The bridge design space has been explored in previous machine learning for design research from both an analytical and empirical perspective [6]. Second, an aircraft wing design example is used to provide an in-depth illustration of the mechanics of the Bayesian approach. These two examples provide well understood and real-world relevant examples to illustrate the background of this design support tool. The third example is provided in the empirical trial. This illustrates how the Bayesian network structure is induced from prior data and the prototype user interface that a designer would use during the design process. This example represents the central application of the Bayesian decision support tool in this paper.

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 necessary and desired features [7] or encoded as an ‘acceptability function’ [8]. For example, a bridge must stretch across a minimum span and be able to support a given load. The span requirement is a necessary feature, while the load requirement can either be encoded as a desirable (support a load of 4 lane traffic is desirable, 2 lane traffic is necessary) or, as the load-bearing of the bridge can be computed, as an acceptability function. The specification guides a 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. For example, there are several alternative approaches to creating a 4 lane bridge, e.g. a cable-stayed bridge, cantilever bridge, etc. This search is frequently supported by using a morphological matrix, where each design variable has a set of possible solution values, of which one value must be selected for each variable.

Conceptual design is by definition fluid. While there is a degree of constraint imposed through the use of morphological matrices, there is ample flexibility during the detail and embodiment stages where the design is crystallised into an artefact that can be manufactured [1]. Specifically, a good concept will be easily transformed in the later design stages into a good final design. Similarly, a poor concept will require extensive effort to be transformed into a good final design, and therefore has a higher likelihood of resulting in a poor final design. For example, consider a bridge across a wide river. A good design concept would be a cable-stayed bridge as this is an efficient structure that can easily span a significant distance. However, it would be possible to elect to design a wooden bridge. Such a bridge would require great care in design to ensure it would be able to meet the specification, and as a result there are many more possibilities for it to fail that specification and hence result in a poor final design. This definition of good/bad concept can frequently only be measured, with whatever metric is appropriate for that product, after the final product has been produced. This 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, approach 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. Such an approach has been adopted for conceptual aircraft optimisation, where a hybrid deterministic and stochastic approach is taken [9,10].

The fluidity of the conceptual design phase means it is difficult to provide concrete evaluation tools. The bridge case study [6] described a partial causal model which was effectively a set of monotonic trends, for example, 'increasing tower height reduces tower stiffness which in turn reduces bridge stiffness'. 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 [11,12]. In effect, these methods aim to provide the most generic design solution that is acceptable. These methods require a pre-defined evaluation function for the design that encodes the original design specification. An alternative stochastically driven method is to bias towards design refinements that do not have 'spiky' probability distribution functions (PDFs) [8]. Such spiky 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 [13], change propagation [14], and other causal graph methodologies [15]. These approaches are based on prior task ordering and then, using the state of design information, to direct the designer to the next suitable task to perform. The approach introduced in this paper, however, uses the shape of the dynamically computed design variable PDFs rather than predefined domain rules to determine the order that the design variables should be determined. This approach has been used for the design and analysis of turbine blades when including probabilistic effects of safety and worst case scenarios [10].

Good concept-stage design models are difficult to obtain. This is because at this stage there typically is a lack of accurate information regarding the interaction between the design variables [16,17]. 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. This research aims to address the algorithmic induction of design models from previous design exemplars, with the specific aim of inducing models that are more easily interpreted by human designers.

The motivating factors for this research are in part the cognitive aspects that affect and constrain human designers [18]. 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. Such stochastic modelling approaches have been attempted in domains where 'natural' uncertainties exist. Vibration analysis and the associated failures of structures under vibration is challenging to compute, and hence a stochastic approach provides a simpler method of predicting failure [19]. In a similar manner, assessing flood risks can not be performed deterministically. By creating a set of simple probabilistic models, it has been possible to assess the risk by combining these simple models [20]. Both these stochastic approaches are founded on building small stochastic partial domain models, and then combining these to provide an overall stochastic domain model. This allows for a designer to rapidly assess risks by only modifying the relevant partial model and then investigating how this affects the global design.

3 Bayesian Design

Uncertainty is a factor in any modelling context. Incorporating this uncertainty in deterministic models is a challenging task. Methodologies have been developed for controlling the uncertainty in early design scenarios [21,22]. These approaches enable designers to use design models that can provide suitable simulation models that are able to provide direction in the design process, albeit with occasionally imperfect predictions. The nature of uncertainty means that it must be modelled stochastically. As described in the previous section, the relationship between the decisions made during the conceptual design stage and the final quality of the product is non-deterministic, complex and uncertain. From an optimisation perspective, this problem is characterised by the lack of high quality numerical models that allow optimisation algorithms to search the design space. An approach to bridge this gap is the use of surrogate or meta-models [17,23,24]. By using a simplified model of a more complete but complex model, it is possible to rapidly search the design space for areas that are more likely to provide promising results. These surrogate models are created by either sampling the design space directly or taking samples from the more complex and costly models.

Bayesian design is the use of Bayesian Belief Networks (BBNs) to support the modelling of the stochastic perspective for the design process. This provides a causal model for a set of design observations or variables [25]. These Bayesian models are represented graphically, where the observations are the graph nodes and the causal links are the directed edges that connect the nodes. As these 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 to 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) PDF. Effectively, the available information biases the unobserved variable's PDF.

This research uses Bayesian ideas to develop a novel design decision support tool. Four core benefits are claimed from implementing a design methodology based around the use of this design support tool:

(1) That a suitably good design domain model can be induced from prior design examples, and this model is useful to designers;
(2) That the design search process can begin with a partial design specification that includes design characteristics;
(3) That the proposed design search heuristics lead to an efficient yet flexible design search path; and
(4) That the resulting designs are at least of equal merit as designs that would arise from a more 'traditional' search method.

These claims will be argued through illustration and empirical evidence. Section 6 provides an illustration of how the initial machine learning process induces the Bayesian design domain model and how this model is then used as part of a dynamic search process. This provides supporting evidence for the first three claims. Section 7 describes the empirical study which provides supporting evidence for the last two claims.

A powerful aspect of using machine learning techniques to induce a design domain model is that the order in which aspects of the design are specified can

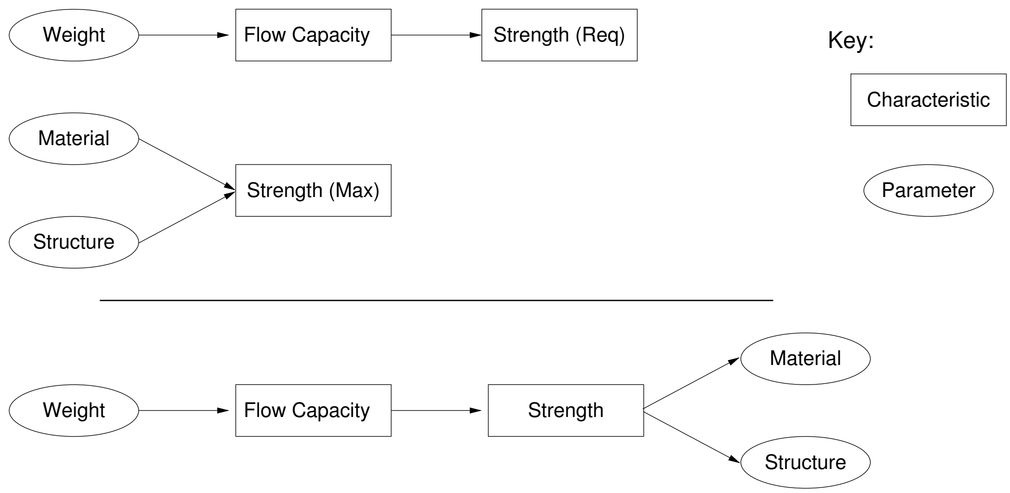


Figure 1. Alternate design models — Top: 'classical' function based approach where design characteristics are functions of design parameters; Bottom: stochastic perspective where design characteristics and perspectives are treated as 'variables'.

be challenged. It is no longer necessary to first specify the design parameters and then either calculate or estimate the resulting design characteristics from these parameter settings. From a Bayesian perspective, all measurable aspects of the designs are treated as 'observed variables'. 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 the values of the design characteristics are a result of the design parameters. For the purposes of this work, this distinction is removed, as it is impossible in general to infer the causal order between the design variables based only on a set of design examples. 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.

The benefits of removing this distinction are illustrated in Figure 1. Under the scheme where design characteristics are seen as 'outputs' of functions of design parameters, the design process must identify design parameter values such that the outputs of independent functions match up. This can be challenging, particularly when the functions in question do not have inverse functions. Using the illustration from the previous paragraph, the outputs are the design characteristics flow capacity and bridge strength (Figure 1). In this simplified case, flow capacity is a function of bridge width, required strength is a function of flow capacity, and bridge strength is a function of material and structural design. For a successful design, bridge strength must exceed the required bridge strength determined by the flow capacity. Depending on which aspects of the bridge are specified, this determines how a designer must proceed: either compute directly the function outputs where possible or identify suitable function inputs to match the desired outputs. Under the stochastic perspective, where the distinction in removed, it is possible to merge the two strength variables so that the matching is done implicitly. Under this scheme, it no longer matters which design variables are specified, as the process to determine the variable values is the same regardless.

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. Under the stochastic view, each design variable has a PDF. These PDFs can either be obtained through sampling previous designs or hand coded [26]. The 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 that 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, the PDFs 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 becomes a much more likely outcome. A schematic illustration is given in Figure 2. This figure represents the distribution function of a design variable that is conditionally dependent on some other design variable. On the left hand function, the parent variable is unspecified and therefore providing no information about how the child distribution will be affected. This unaffected child distribution is quite 'wide', suggesting that for this variable, most settings are of equal likelihood. When the parent variable is set, this alters the distribution function of the child, in this case narrowing it down towards the higher values of the child variable state. The use of design variable based distribution functions in design has been reported in the design analysis of heat transfer in turbine blades enabling a designer to design for specific reliability [10], project planning and management by computing the probability distributions of individual task lead times [27], and decision making for scientific investment strategies based on distributions of risks associated with various

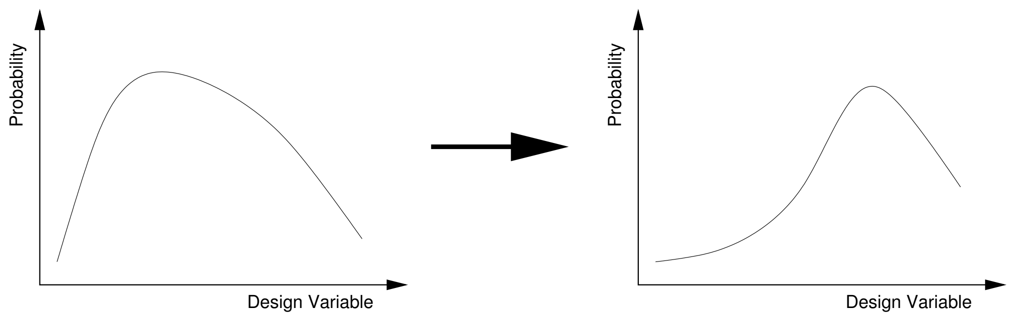


Figure 2. Schematic illustration of how a design variable's PDF changes when the parent variable is specified. The function on the left is the marginal distribution and the right represents the updated distribution given further information.

alternative options [28].

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. However, it will be assumed that the designer wishes to specify a design concept that has the greatest probability of resulting in a good product, and therefore be a concept that requires the least 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 the variables with the 'flattest' PDFs being last. As a PDF must sum to unity, a simple approach is to consider the difference between the PDF's maximum and minimum values. The greater this difference is, the 'spikier' the PDF is. This heuristic 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, in a manner similar to the methodology proposed to minimise the effects due to change propogation [14]. 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.

3.1 Qualitative illustration

A hypothetical wing design example is used to provide a qualitative illustration of the Bayesian design methodology. Wing design is a well understood domain with known deterministic models. Hence, this does not represent a typical application domain for the Bayesian tool, but provides a context in which insights into the underlying algorithmic mechanisms of the Bayesian design decision support tool can be made. Further, in this illustrative case the model structure is assumed to be known *a priori*. Under the full application of this methodology, the model structure would be induced from prior design data. Finally, by modelling the wing domain with three design variables, this provides the simplest possible non-trivial BBN structure.

This wing design domain will be defined by one design parameter: wing geom-

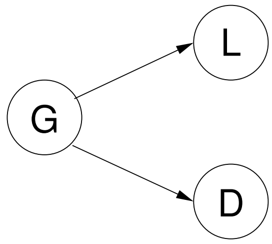


Figure 3. Bayesian Belief Network for hypothetical wing design (G = wing geometry, L = lift, and D = drag).

etry (effectively representing chord thickness), and two design characteristics: wing lift and wing drag. Figure 3 illustrates the causal relationship between these design variables: namely that the wing geometry determines both the lift and drag factors. For simplicity in this hypothetical example, it is assumed that lift and drag are independent of each other.

Two design scenarios are illustrated. The first is where the ‘parent’ variable is known, in this case the wing geometry. This is used to then determine estimates on the wing lift and drag factors. The second scenario starts with a known ‘child’ variable, in this case the wing lift factor. This scenario illustrates how the Bayesian design then directs the designer to determine the wing geometry and place an estimate on the lift factor.

The underlying domain rules are illustrated in Figure 4. There are two basic domain rules, one for each of the design characteristics determined by the value of the design parameter, wing geometry in this case. These rules are summarised by: as wing geometry increases, both lift and drag factors increase. However, nominal lift increases at a diminishing rate but the spread of possible lift values widens. Conversely, nominal drag increases at an accelerating rate while the spread of possible drag levels reduces. It is assumed that these domain rules are not explicitly known by the designer, and hence that the designer is strictly using the Bayesian support tool for guidance. However,

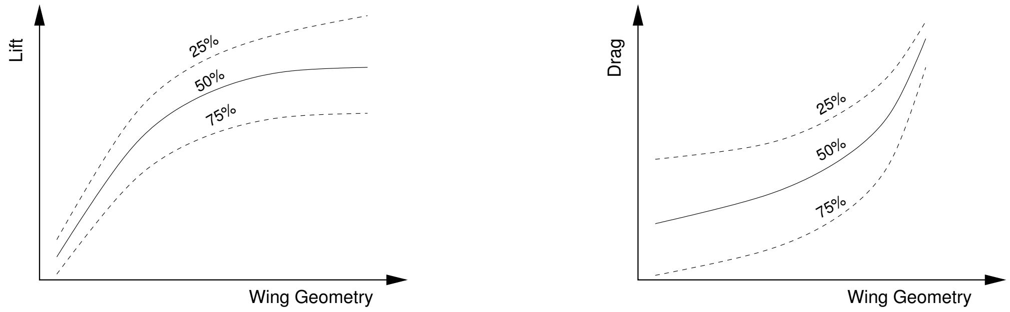


Figure 4. Rules governing the hypothetical wing design, with contours (dashes) representing the distributions from the nominal rules (solid).

by providing these domain rules explicitly in this illustration, it is possible to better understand the nature of the Bayesian design support tool's output.

3.1.1 Specified Parent Variable

In the first design scenario, the wing geometry has been predetermined to accommodate the various flight control systems and fuel storage. The wing geometry has been set to 'low'. The designer wishes to determine what lift is possible to be obtained from this wing and what drag is to be expected from this design.

The designer enters the low value for the wing geometry into the design support tool. This then provides the conditional PDF's given by:

$$f_L(x) = \mathrm{P}(L = x \mid G = \text{low}) \quad (1)$$
$$f_D(x) = \mathrm{P}(D = x \mid G = \text{low}) \quad (2)$$

These distributions are obtained by considering the distributions presented in Figure 4 at the given wing geometry value, effectively projecting the distribution from this geometry value. The projected distributions are presented in Figure 5. From here it can be seen that the 'lift' distribution is 'spikier'

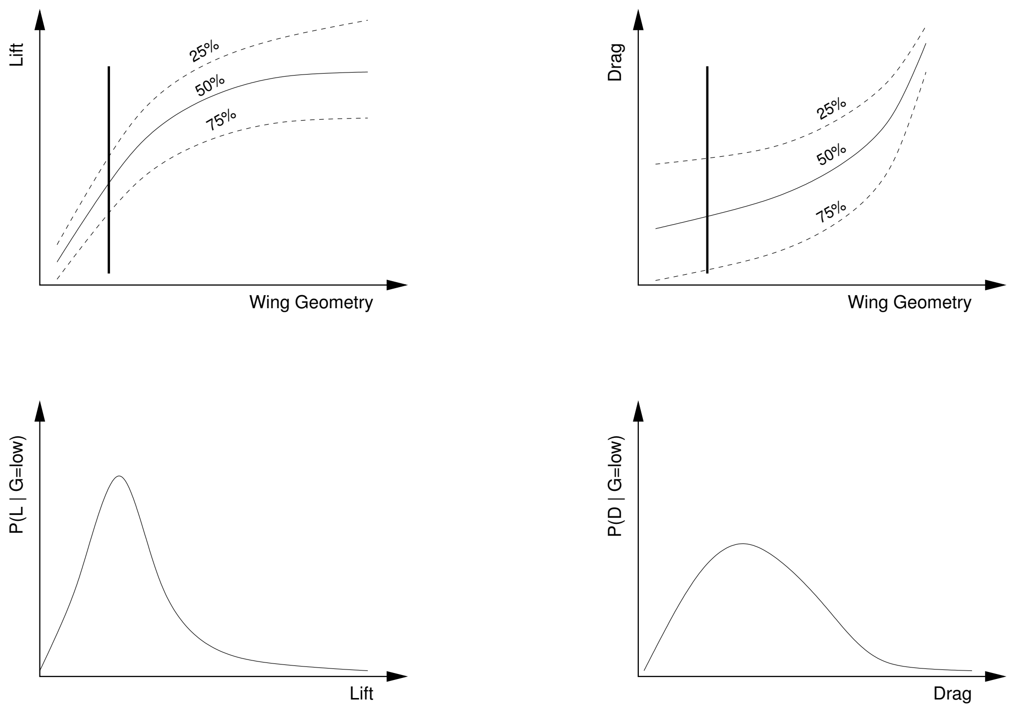


Figure 5. The projection of the domain rules onto the conditional probability distribution function, given a specified geometry (heavy vertical lines on top plots).

than the 'drag' distribution, and hence the design heuristic is to determine the lift variable first. The distribution suggests that the designer should expect a lower lift value for the given wing geometry. The designer is at liberty to select a high lift value, but this would be expected to be significantly more difficult to fulfil in the later design stages. As drag is independent of lift (under this domain model), the setting of the lift value does not affect the drag distribution. The designer is at liberty to select the drag target for this design, but is suggested to aim for the mode of the distribution.

3.1.2 Specified Child Variable

In the second design scenario, the design specifies that the wing is required to provide a high lift factor. Hence, the wing lift factor has been set to 'high', and the designer wishes to determine what wing geometry would achieve this lift and obtain an estimate on the expected drag factor for this design.

The designer enters the high value for the lift factor into the design support tool. In this case, as the 'lift' design variable is a child variable in the model, the distribution function for the wing geometry represents a likelihood distribution rather than a probability distribution. The distinction between these is that the integral of the likelihood function can be greater than unity, which is the case when a number of wing geometries are highly likely to provide the specified lift. However, the principles behind using the distributions as a support tool for selecting variable values remains the same: the designer is encouraged to select the value where the distribution peaks, as this represents the most likely outcome given the currently specified design.

The distribution functions in this case are given by (see Figure 6):

$$f_G(x) = \mathrm{P}(L = \text{high} \mid G = x) \tag{3}$$

$$f_D(x) = \mathrm{P}(D = x) \tag{4}$$

It is worth noting that the distribution function for the drag is simply the *a priori* drag distribution. This is as the domain model in this hypothetical example has lift and drag as independent variables. Therefore, with only lift being set, there is no relevant information with respect to drag and hence the baseline drag distribution results as the drag PDF.

The designer uses the 'spiky distribution first' heuristic to determine which variable should be set next. In this case, the drag distribution is the spikier and hence it is determined next. As drag is not part of the specification, the designer is directed to selecting the mode value of the drag distribution. However, the designer is at liberty to use their engineering judgement to select a drag value slightly below the mode, as this provides for a more efficient final

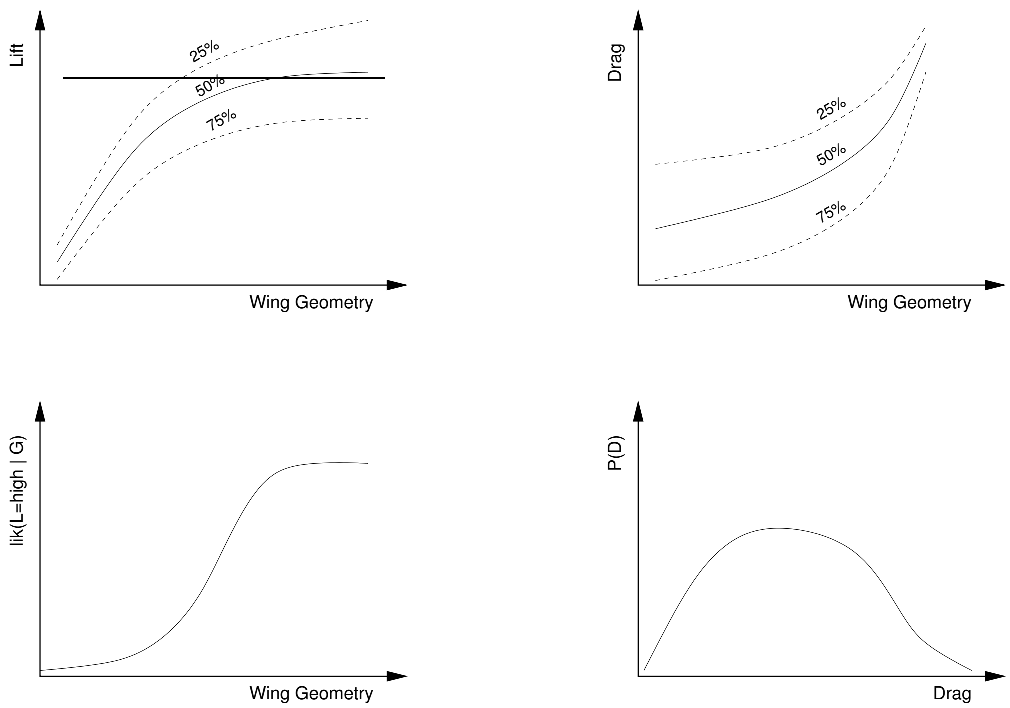


Figure 6. The likelihood distribution for wing geometry, given that the wing lift has been specified. Note that the drag distribution function remains the baseline distribution.

wing design.

The wing likelihood distribution now needs to be recomputed, as it is affected by the additional information given by setting the drag variable. This distribution is given by (see Figure 7):

$$f'_G(x) = \mathrm{P}(L = \text{high},\ D = \text{low} \,|\, G = x) \tag{5}$$

The geometry likelihood distribution has now been narrowed from the distribution in the previous iteration. Again, the variable value determination heuristic suggests that the design should set the value to the distribution's mode, in this case a 'middle' value for wing geometry. This finalises the design, and provides a realistic design specification to be taken forward to the next design stage.

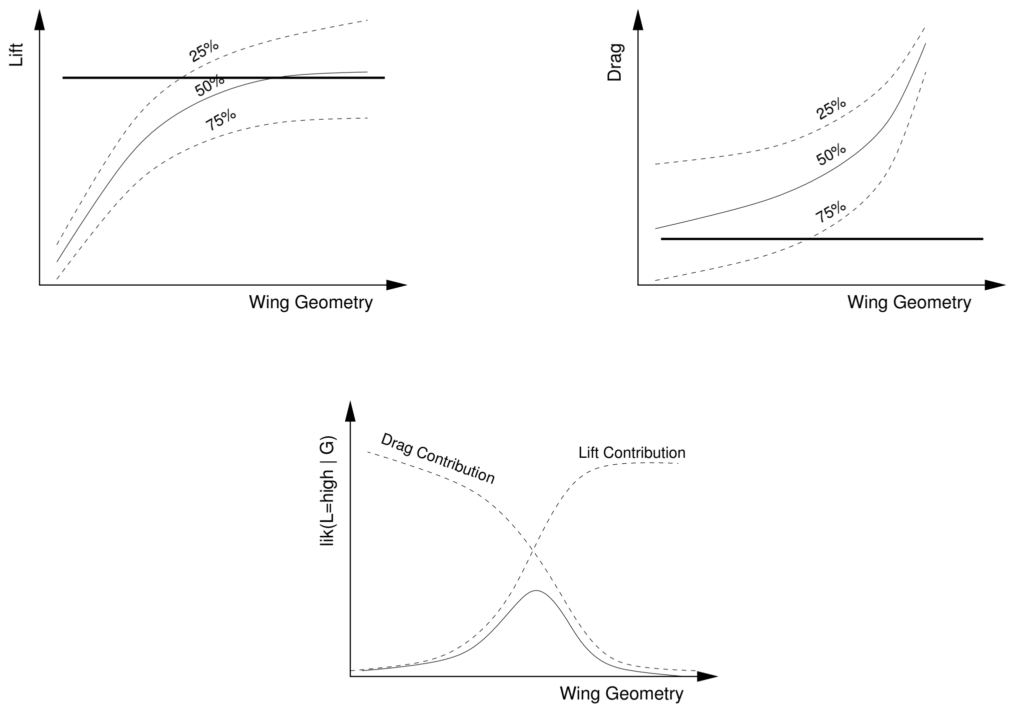


Figure 7. The likelihood distribution for the wing geometry where both lift and drag have been specified (solid line). The contributions for both lift and drag are illustrated (dashed lines).

4 Inducing Bayesian Networks

To use BBNs as a design support tool, it is essential to acquire a sufficiently accurate 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 [29]. 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, in terms of number of potential arcs, 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 significantly greater consideration than identifying the causal links. As a result, the expert crafted BBN is not appealing.

This paper develops an alternative method for identifying the causal links in the BBN by applying data mining techniques to a database of previous design exemplars. This data mining algorithm analyses the given database and creates a network that provides a sufficiently close representation of the stochastic phenomenon observed in the database. Data mining algorithms use three main metrics to determine model quality: *validity*, *understandability*, and *interestingness* [30]. 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

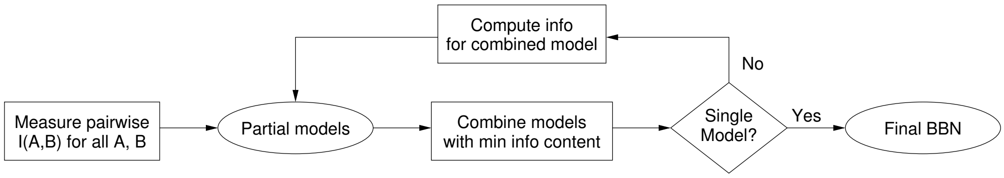


Figure 8. Flowchart representing the greedy BBN learning algorithm.

not known, the complexity of most BBN graph inducing algorithms explodes to $O(n!)$, where n is the number of domain variables. In this research, it is assumed that the causal order of the design variables is not known prior to running the algorithm. In terms of computational resources, this places a significant constraint for inducing BBN graphs for realistic design problems.

Information content has been used in engineering applications ranging from arguments for basing a design theory [31] through to enhancing the diagnostic and maintenance process [32]. For the purposes of this research, the information content approach enables a significant reduction in the computational complexity involved in learning graphical models. Chen et al. [33] describe a graph inducing algorithm based on measuring the conditional independence between pairs of variables. Where the causal order of the variables is not known, they show that the graph can be learnt with $O(n^4)$ complexity. In this research, this idea has been adapted to create a novel greedy algorithm that further reduced the computational complexity down to $O(n^2)$. This 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 8.

The graph search algorithm developed in this research implements a breadth-first greedy search heuristic based on a measure of the information content of the conditional probability distribution. Recall the definition of conditional probability:

$$P(B = b \mid A = a) = \frac{P(B = b, A = a)}{P(A = a)} \tag{6}$$

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 variance in this difference:

$$I(A, B) = \mathbf{E}[P(B \mid A) - P(B)]^2 \tag{7}$$

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 reasonably 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 models with the lowest information content scores and a shared variable are merged, resulting in one fewer partial model. 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 empirically test the design heuristics described in Section 4, it was necessary to implement the stochastic algorithms: one algorithm to induce the domain model and another to implement the interactive design search. 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 7 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 datasets, this was not a part of the user trial. The second part of the code represents the user interface to the BBN. Figure 9 contains the flowchart for the iterative and designer led search process. This was 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

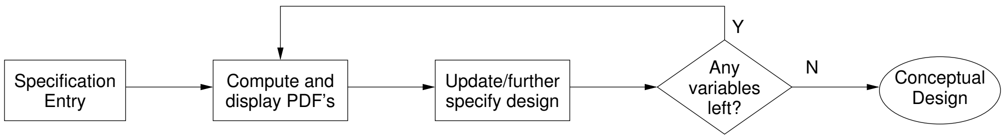


Figure 9. Flowchart representing the overall design search process.

on a separate worksheet. 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.

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 interested only 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 (see Figure 12 for a screen shot). 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 $\mathrm{P}(X|\pi(X))$.

Finally, the dataset worksheet 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 centred 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 $\mathrm{P}(Y = y|\pi(Y))$ 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. Thus, 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 a 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:

$$\mathrm{P}(X|Y = y, \pi(X)) \tag{8}$$

There are now two further sub-cases to consider: X has been specified and X has not been specified. Where X has been specified, 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 8 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 likelihood 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 two 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 (Section 3), the first heuristic that guides the order in which variables are determined suggests that the designer focuses initially on the variables with narrow distributions and then moves onto variables with ever wider distributions. 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 Empirical study domain: Preliminary Car Design

For the purposes of illustrating and arguing the benefits of the Bayesian design support tool, a simple conceptual car design domain is used. This design example will be used in two forms: the first will be to provide an illustration of how the dynamic search support tool operates (this section), and will provide evidence for Claims 1–3 (recall Section 3). The second form (Section 7) will be the basis for the empirical trial which provides empirical evidence supporting further evidence for Claim 3 and evidence for Claim 4.

For this laboratory based trial, the well known machine learning car design database was used [34]. This database contains a sample of 1728 fully described designs. The design domain is defined by ten variables, of which six are design parameters (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) and the remaining four are design characteristics (the overall cost of ownership; the comfort level; the technology level; and the overall car acceptability). All the variables are discrete, and hence this fully defines the domain's morphological matrix, and a more detailed description of the variables is listed in Table 1. The original car design database was constructed using a set of predetermined rules. The structure of these rules is provided in Figure 10. This known rule structure makes it possible to evaluate the quality of the machine learnt domain model.

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 identifies 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 these 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

Table 1
Car design morphology table: variable names, abbreviations, and description. The descriptions include the possible variable values. Design parameters are in lower case and the design characteristics are in upper case.

Name (abbreviation)	Description
buying (buy)	Purchase price for car (low, medium, high, very high)
maintenance (mnt)	Expected maintenance cost for car (low, medium, high, very high)
doors (drs)	Number of doors on car (2, 3, 4, 5+)
persons (pers)	Number of passengers (2, 4, 5+)
luggage (lug)	Available luggage volume (small, medium, big)
safety (safe)	Designed safety level (low, medium, high)
COMFORT (CMFT)	Comfort level of car (unacceptable, acceptable, good, very good)
PRICE (PRC)	Total cost of ownership (unacceptable, acceptable, good, very good)
TECHNOLOGY (TECH)	Technology level of car (unacceptable, acceptable, good, very good)
CAR (CAR)	Overall acceptability of car (unacceptable, acceptable, good, very good)

illustrated in Figure 10. As the model induction algorithm produced an exact replica of the original rule structure, this provides evidence for Claim 1. Subsequently, this network was encoded in the Excel spreadsheet, along with the design database.

6.1 Illustration of a Stochastic search

The Excel spreadsheet, coupled with a Visual Basic macro, provides the 'user interface' to the stochastic design tool as shown in Figure 11. For illustration purposes, one of the empirical design scenarios is described in this section. This illustrates Claim 2: how the design is initially partially specified, and

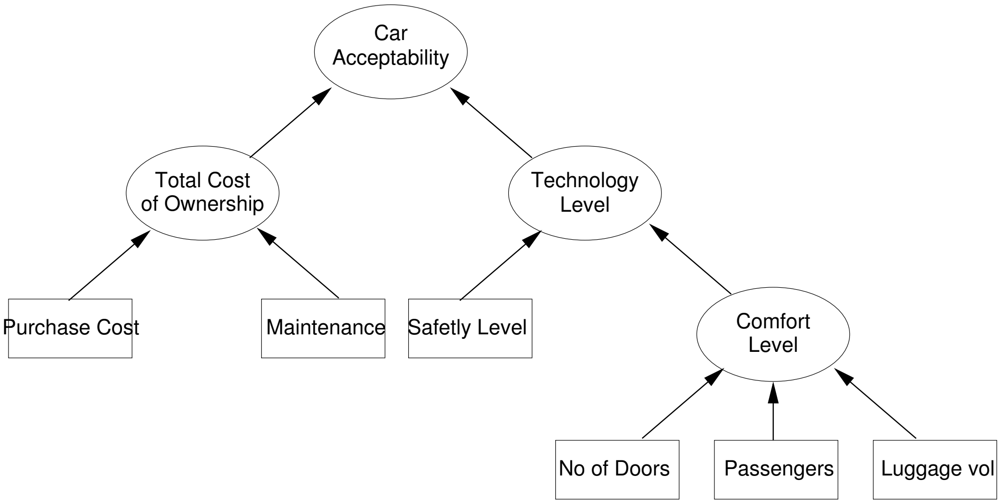


Figure 10. Rule structure for the conceptual car domain.

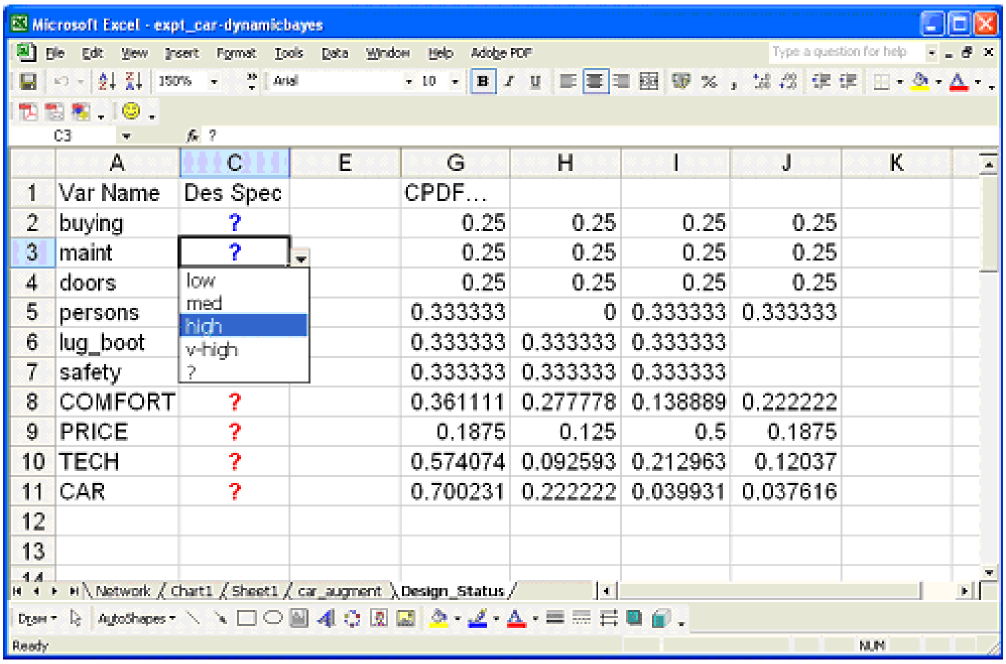


Figure 11. Screen shot from the design support tool, prior to any specified design variables.

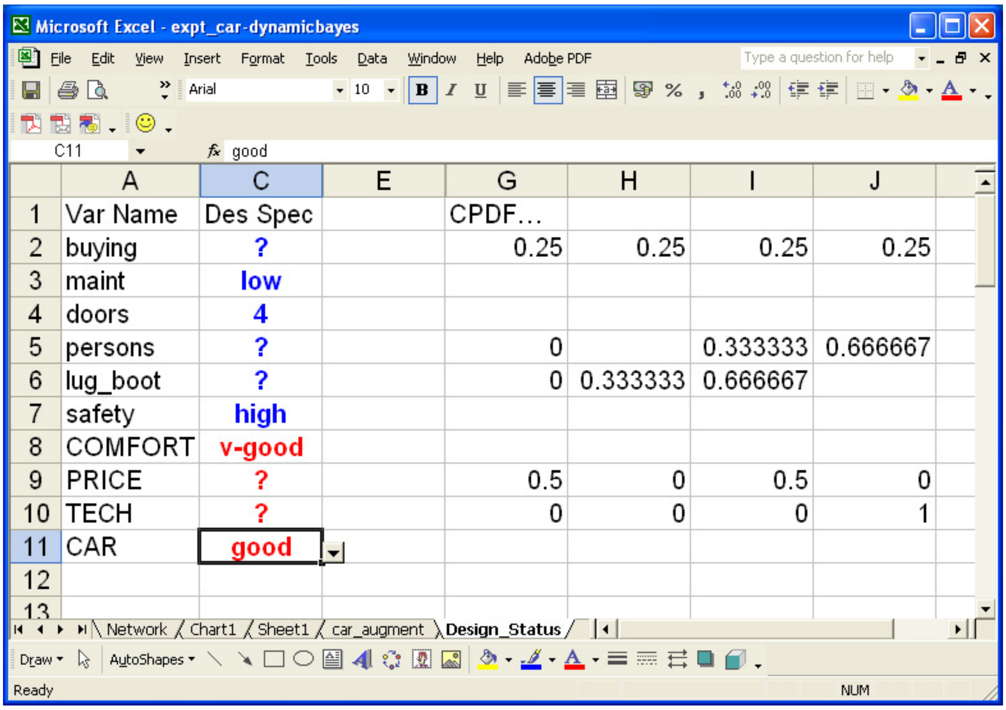
how the stochastic search heuristics guide the designer to further refine the design specification.

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 'very good' comfort level and it should have a 'good' acceptability level. This specification was translated into design variable settings, and entered into the Excel spreadsheet. Figure 12 is a screen shot taken after the initial design specification was entered, and the VB macro run.

Table 2 contains the output of the stochastic search tool for each subsequent step after the initial specification was entered. For each step the likelihood of each remaining open variable setting is displayed. In the table, the variable name in boldface represents the variable that was determined at that step. The value it was set to is similarly in boldface. To summarise, the stochastic search method suggested the following course of action:

(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 four occasions where the guidance to selecting the variable value was ambiguous. For example, determining the overall cost of ownership ('PRICE') placed equal weight between selecting 'low' or 'high' (see Step 3 in Table 2). In this case, as the car is intended to be 'accessible', so 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'. Overall, this represents an efficient and flexible search process, illustrating Claim 3.



	A	C	E	G	H	I	J
1	Var Name	Des Spec		CPDF...			
2	buying	?		0.25	0.25	0.25	0.25
3	maint	low					
4	doors	4					
5	persons	?		0		0.333333	0.666667
6	lug_boot	?		0	0.333333	0.666667	
7	safety	high					
8	COMFORT	v-good					
9	PRICE	?		0.5	0	0.5	0
10	TECH	?		0	0	0	1
11	CAR	good					

Figure 12. Screen shot from the 'Accessible Luxury' design specification and initial PDF computation.

6.2 'Traditional' design 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 available immediately 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.

This approach was implemented using a similar user interface. Here, the subjects could specify only the design parameters. When sufficient information

Table 2
Search path for the unspecified design variables for the 'Accessible luxury'. Selected variable/value typeset 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	
	PRICE	0.5	0	0.5	0
	TECH	0	0	0	**1**
2	buying	0.25	0.25	0.25	0.25
	persons	0	0.33	0.67	
	luggage	0	0.33	**0.67**	
	PRICE	0.5	0	0.5	0
3	buying	0.25	0.25	0.25	0.25
	persons	0	1	1	
	PRICE	**0.5**	0	0.5	0
4	buying	1	1	0	0
	persons	0	**1**	1	
5	**buying**	**1**	1	0	0

was available, the interface reported the computable design characteristics. The subjects still had the rule structure network to hand (Figure 10), and therefore could use this as a limited guide to the design search process.

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 little or no 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.

6.3 Summary of the effects of Bayesian design

The car design case study provided in this section illustrates the first three of the four claims made in Section 3. The first claim is that the machine learning algorithm induces a suitably good domain model from a set of prior design examples. The algorithm produced a graph structure with one arc too many for it to be a tree structure, as required for a BBN. By using the information content heuristic, it was possible to identify which arc should be deleted and this resulted in the same structure that generated the data in the first instance.

The second claim is that the design search process can begin with a partial design specification. This was demonstrated by starting the design search with a specification on a subset of both design parameters and characteristics. The interactive search tool then guided the design refinement process, variable by variable. For each variable, the various possible settings were ordered according to the probability of a successful outcome. The designer is encouraged to follow this 'path of greatest likelihood', but is not compelled to. The illustration of this search process thus provides evidence for the third claim, namely that the design search heuristics lead to an efficient yet flexible design search path.

7 Empirical comparison of two design tools

The stochastic design tool was compared to the 'traditional' design approach through experiment. In addition to measuring the time taken on each design task and analysing these designs, it was also possible to observe the different search tactics that emerged from both approaches. The design merit was determined by whether the design fulfilled the specification in each case. This comparison between the stochastic design tool and a more traditional approach provides empirical evidence supporting Claims 3 and 4 from Section 3.

7.1 Methodology

Fundamentally, the experiment tested for differences between the design search time and the nature of the final designs arising from the two different search tools. Four different design specifications (or scenarios) were provided with the following characteristics: (1) only design parameters specified ('people carrier'); (2) only design characteristics specified ('sports car'); (3) both design parameters and characteristics specified ('accessible luxury'); and (4) an infeasible specification ('imported car'). The final, 'infeasible' specification was designed to test the ease of identifying a suitable modification to the specification so that it became feasible.

The subjects were randomly allocated one of the two design search tools. After a briefing on the experiment, they were allowed 45 minutes to complete and record the four designs. The designs were recorded using a paper-based form. Throughout the experiment, the researcher was available to resolve any queries. The researcher was also in a position to observe directly the use of the two different design tools.

On completion of the experiment, the design and time data were collated for statistical analysis. This provided a population-based comparison approach for the two sources of designs. The hypothesis under test was that the results of the two different design tools were samples from two different populations against the null hypothesis being that the two different sets of designs were from the same population.

7.2 Execution

The design subjects were taken from the third year of a four year Master's of Engineering degree course and randomly divided into two groups. Prior to the design work, the subjects were given an overview of both approaches and a summary of the design context. This material was also provided as a handout, should further reference to the material have been needed. In addition, the researcher was available for support if required.

Each subject was provided with a computer running the appropriate version of the design search tool. They were instructed to search the design space for each specification until they were satisfied with their conceptual design. This design was then recorded on the paper forms by noting down the values for each of the design variables. In addition, the subjects were asked to record start and finish times.

Throughout the search process, observations were made on the different usage of the two design search tools. This was primarily to evaluate the effectiveness of the user interface.

The data collected from the 17 design subjects was grouped according to the four design tasks. There were nine subjects using the traditional ('rule-based') design search tool and eight subjects using the stochastic search tool. By considering the means, standard deviations, and population sizes for each design variable in each task, a two tailed t-test was used to determine the probability that the two sets of design variable observations came from the same population [35].

The results are presented in Table 3 grouped by design task. For each design task, the variables are annotated on whether they were part of the specification or not along with the t-test probability that the samples from the two different design tools arose from the same population.

As the trial used small samples ($n_{\mathrm{Bayes}} = 8$ and $n_{\mathrm{Trad}} = 9$), it shall be assumed that the 10% significance level provides evidence that the observations occur from different populations. Where this is the case, these have been highlighted in Table 3 using boldface. The entries where there is strong evidence both for ($t \leq 10\%$) and against ($t \geq 10\%$) the hypothesis have been highlighted.

Although there is not conclusive evidence that most variable settings either belong to the same sample population or not, all designs that were produced were of similar merit in the sense that they met the specification (with the exception of 'Imported Car', which represented an unfeasible design specification). Thus, it can be concluded that the designs produced using this prototype stochastic design search tool are of at least equal merit to those produced using a more traditional method and hence providing empirical evidence to Claims 3

Table 3
Results from the t-test analysis ($n_{\text{Bayes}} = 8$ and $n_{\text{Trad}} = 9$). For each design task, the top row denotes with a check ($\surd$) if the variable was part of the design specification. The t-values in boldface represent supporting evidence at the 10% level: $t \leq 10\%$ indicates samples arose from different populations and $t \geq 90\%$ indicates samples arose from the same population.

	buy	mnt	drs	pers	lug	safe	CMFT	PCE	TECH	CAR
	People Carrier									
Spec		$\surd$		$\surd$	$\surd$	$\surd$				
t (%)	41	**93**	**0.5**	**100**	**10**	30	**0.6**	38	36	**10**
	Sports Car									
Spec							$\surd$	$\surd$	$\surd$	
t (%)	32	67	83	**0.6**	36	48	**6**	73	31	67
	Accessible Luxury									
Spec		$\surd$	$\surd$			$\surd$	$\surd$			$\surd$
t (%)	19	15	12	12	27	30	**8**	**5**	**92**	**7**
	Imported Car									
Spec			$\surd$	$\surd$	$\surd$	$\surd$	$\surd$		$\surd$	$\surd$
t (%)	**91**	56	**100**	**100**	57	81	**9**	39	11	75

and 4.

7.4 *Model analysis*

A final note must be made regarding the nature of the stochastic model. In Section 4, three main data mining algorithm metrics were referred to. These measure an algorithm's ability to produce models with validity, understandability, and interestingness [30]. No formal definitions were provided for these metrics, but they serve as useful guidelines for assessment. In this case, the validity of the model is clear: the algorithm reproduced the source model and hence performed well on this measure. Understandability and interestingness are more subjective metrics. The model produced is a simple tree structure, which indicates a relatively easy model to understand as there is a clear and tractable path that determines the affect of local changes on the global design state. The final metric, interestingness, is not possible to evaluate for this design domain, due to the artificial nature of this test domain.

8 Discussion

The t-test analysis results in Table 3 does not provide strong evidence either for accepting or rejecting the original hypothesis that there is a difference in the resulting designs using this stochastic search tool. In each design task, there is evidence supporting both options. It would be expected that the design variables in each specification would arise from the same population, however it can be seen that this conjecture is both strongly supported and rejected. In Table 3 an example of this is provided in the 'People Carrier' design, where both 'maintenance' and 'luggage' form part of the specification, but the t-test values are at opposite ends of the spectrum. To a lesser extent, this would also be expected in the remaining, designer determined, design variables. In addition, for a large number of the samples, the t-test does not provide strong support that the populations are from the same or different populations. This final issue is in part due to the small sample size that is typical in subject-based design research.

Another possible reason for the lack of clear differentiation, or otherwise, of the two samples arises from the nature of the design problem. The conceptual car problem was selected due to the availability of data and previous analytic work thereon in the machine learning domain. The car design domain is largely intuitive, that is, the design model is roughly aligned with what an informed engineer would expect of a car design model (for example, increasing the number of doors on a car results in an increased comfort level). It was not possible to measure the effect this had on either search tool.

The designers reported little problem with the final design task, the infeasible design specification. This was not surprising, as these design students were able to rapidly identify that this design task was infeasible and were prepared to challenge and modify the original specification to be able to produce a 'workable' design concept.

From observations made while the designers were searching the design space, some interesting search behaviour emerged. In the traditional, rule-based, search method designers would complete all the design parameters so that the design characteristics would all be evaluated. This design would typically not meet all the requirements, and so the designer would need to modify the design parameters. This modification process would typically involve the designer 'hunting' around the first attempt by rapidly testing different parameter settings and noting their effect on the design characteristics. The designers using the stochastic search tool also developed a hunting method, however rather than observing the direct affect of changing a design variable, these designers would review the global change on the design space. This enabled the stochastic-based designers to gather more information and better consider their decisions, however this was at the cost of requiring more time per design.

9 Conclusions

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. As such, this provides evidence of good model validity, as discussed in Section 3.

Using the BBN induced from the design database as a dynamic morphology 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. This would support the measuring of the model understandability.

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. All designers referred to the rule structure diagram to identify the 'neighbouring' design variables for modification. However, no guidance was provided on which would be the best variable to modify in a particular situation. An improvement in the stochastic search tool would be to provide some form of guidance to identify fruitful modifications to the current partial over-constrained design specification. This would allow the designer to 'unblock' the infeasible design specification using a minimal change to the original specification. Investigating this aspect will begin to suggest means for identifying the interestingness of the models.

Using the BBNs 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 morphological design space in other sequences. This offers the designer the flexibility that is essential during the conceptual design stage.

10 Future work

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 a metric 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. Alternatively, designers could benefit from better understanding of the meaning of the PDF profiles. It is not clear how much experience is needed with this design data feedback method before a designer can rapidly interpret the presented information.

The stochastic design search tool was compared to a rule-based tool that would rapidly evaluate designs as they were created. This is not a realistic situation, and provided a challenging datum to compare the stochastic search tool against. Further experimentation with more realistic conceptual evaluation tools should be performed.

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. This should also involve a less ‘intuitive’ case, where a typical designer would not be expected to have little prior expectation on how the design domain is structured or behaves.

Acknowledgements

This research is funded by a Nuffield Foundation Award to Newly Appointed Lecturers in Science, Engineering and Mathematics (Grant number: NAL/008-46/G). Thanks must also be given to the design students who provided their time to support the empirical work.

References

[1] G. Pahl, W. Beitz, Engineering Design: A Systematic Approach, 2nd Edition, Springer-Verlag London, 1996.

[2] N. Cross, Engineering Design Methods: Strategies for Product Design, John

Wiley, Chichester, UK, 1994.

[3] K. T. Ulrich, S. D. Eppinger, Product Design and Development, 2nd Edition, McGraw-Hill, Boston, MA, 2000.

[4] B. Hollins, S. Pugh, Successful Product Design, Butterworths, 1990.

[5] F. Zwicky, The morphological approach to discovery, invention, research and construction, in: F. Zwicky, A. G. Wilson (Eds.), New Methods of Thought and Procedure, Springer-Verlag, 1967.

[6] Y. Reich, Design knowledge acquisition: Task analysis and a partial implementation, Knowledge Acquisition 3 (3) (1991) 237–254.

[7] S. Pugh, Total Design, Addison-Wesley, 1990.

[8] D. R. Wallace, M. J. Jakiela, W. C. Flowers, Design search under probabilistic specification using genetic algorithms, Computer Aided Design 28 (5) (1996) 405–421.

[9] G. Lomdardi, G. Mengali, F. Beux, A hybrid genetic based optimization procedure for aircraft conceptual analysis, Optimization and Engineering 7 (2) (2006) 151–171.

[10] R. S. R. Gorla, Probabilistic heat transfer and structural analysis of a turbine blade, International Journal of Turbo and Jet-engines 22 (1) (2005) 1–11.

[11] G. Taguchi, E. A. Elsayed, T. C. Hsiang, Quality Engineering in Production Systems, McGraw-Hill, 1989.

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

[13] P. J. Clarkson, J. R. Hamilton, Signposting: a parameter-driven task-based model of the design process, Research in Engineering Design 12 (1) (2000) 18–38.

[14] P. J. Clarkson, C. Simons, C. M. Eckert, Predicting change propagation in complex design, Journal of Mechanical Design 126 (5) (2004) 788–797.

[15] S. Gentil, J. Montmain, Hierarchical representation of complex systems for supporting human decision making, Advanced Engineering Informatics 18 (3) (2004) 143–159.

[16] J. E. Pacheco, C. H. Amon, S. Finger, Bayesian surrogates applied to conceptual stages of the engineering design process, Journal of Mechanical Design 125 (4) (2003) 664–672.

[17] N. Kovac, I. Sarajcev, D. Poijak, A numerical-stochastic technique for undergraound cable system design, IEE Proceedings: Generation, Transmission and Distribution 153 (2) (2006) 181–186.

[18] G. A. Miller, The magical number seven, plus or minus two: some limits on our capacity for processing information, Psychological Review 63 (1956) 81–97.

[19] L. Katafygiotis, S. H. Cheung, Domain decomposition method for calculating the failure probability of linear dynamic systems subjected to Gaussian stochastic loads, ASCE Journal of Engineering Mechanics 132 (5) (2006) 475–486.

[20] H. Apel, A. H. Thieken, B. Merz, G. Bloschl, A probabilistic modelling system for assessing flood risks, Natural Hazards 38 (1–2) (2006) 79–100.

[21] N. V. Sahinidis, Optimization under uncertainty: State-of-the art and opportunities, Computers & Chemical Engineering 28 (6–7) (2004) 971–983.

[22] D. J. Pons, J. K. Raine, Simulation of key performance characteristics under uncertainty, Proceedings of the Institution of Mechanical Engineers Part B: Journal of Engineering Manufacture 219 (1) (2005) 151–162.

[23] I. G. Osio, C. H. Amon, An engineering design methodology with multistage Bayesian surrogates and optimal sampling, Research in Engineering Design 8 (4)

(1996) 189–206.

[24] V. C. P. Chen, K. L. Tsui, R. R. Barton, M. Meckesheimer, A review on design, modelling and applications of computer experiments, IIE Transactions 38 (4) (2006) 273–291.

[25] F. V. Jensen, Bayesian Networks and Decision Graphs, Statistics for Engineering and Information Science, Springer, New York, NY, 2001.

[26] B. H. Thacker, D. S. Riha, S. H. K. Fitch, L. J. Huyse, J. B. Pleming, Probabilistic engineering analysis using the NESSUS software, Structural Safety 28 (1–2) (2006) 83–107.

[27] S. H. Cho, S. D. Eppinger, A simulation-based process model for managing complex design projects, IEEE Transactions on Engineering Management 52 (3) (2005) 316–328.

[28] S. S. Farid, J. Washbrook, N. J. Titchener-Hooker, Combining multiple quantitative and qualitative goals when assessing biomanufacturing strategies under uncertainty, Biotechnology Progress 21 (4) (2005) 1183–1191.

[29] T. N. S. Murdoch, N. R. Ball, Machine learning in configuration design, Artificial Intelligence for Engineering Design, Analysis and Manufacturing 10 (1996) 101–113.

[30] S. Mitra, S. K. Pal, Data mining in soft computing framework: A survey, IEEE Transactions on Neural Networks 13 (1) (2002) 3–14.

[31] N. P. Suh, The Principles of Design, Oxford Series on Advanced Manufacturing, Oxford University Press, Oxford, UK, 1990.

[32] L. Qu, L. Li, J. Lee, Enhanced diagnostic certainty using information entropy theory, Advanced Engineering Informatics 17 (3–4) (2003) 141–150.

[33] J. Chen, R. Greiner, J. Kelly, D. Bell, W. Liu, Learning bayesian networks from data: An information-theory based approach, Artificial Intelligence 137 (1–2) (2002) 43–90.

[34] C. L. Blake, C. J. Merz, UCI repository of machine learning databases, `http://www.ics.uci.edu/~mlearn/MLRepository.html` (1998).

[35] E. Kreyszig, Advanced Engineering Mathematics, eighth Edition, Wiley, New York, NY, 1999.