A knowledge-based system for numerical design of experiments processes in mechanical engineering

Gaëtan Blondet[a], Julien Le Duigou[b]*, Nassim Boudaoud[b]

[a] Phimeca Engineering, 18 Boulevard de Reuilly, 75012 Paris, France
[b] Sorbonne Universités, Université de Technologie de Compiègne, Mechanical laboratory Roberval FRE UTC/CNRS 2012, CS 60319, 60203 Compiègne, France
*Corresponding author

Email addresses : blondet@phimeca.com (G. Blondet), Julien.le-duigou@utc.fr (J. Le Duigou), Nassim.boudaoud@utc.fr (N. Boudaoud).

CRediT author statement:

Gaëtan Blondet: Methodology, Software, Investigation, Data Curation, Resources, Writing – Original Draft, Visualization. **Julien Le Duigou**: Conceptualization, Methodology, Writing – Review & Editing, Supervision, Project Administration, Funding Acquisition. **Nassim Boudaoud**: Conceptualization, Methodology, Validation, Writing – Review & Editing, Supervision.

Highlights

- An architecture for numerical design of experiment configuration is proposed.
- A bayesian network with a “multi-net” strategy is proposed.
- Models are trained from historical data and expert knowledge.
- The performances of the proposed method are validated through a case study.

Abstract

This paper describes a specific Knowledge-Based System (KBS) to assist designers in configuring Numerical Design of Experiments (NDoE) processes efficiently. NDoE processes are applied in product design to improve the quality of product, by taking into account variabilities and uncertainties. NDoE processes are defined by various and complex methodologies to achieve several objectives, as optimization, surrogate modeling or sensitivity analysis. On the other hand, NDoE processes may demand huge computing resources to execute hundreds simulations, and also advanced expert knowledge to set the best configuration amongst numerous possibilities. Designers aim to obtain most useful results with a minimal computational cost as soon as possible. Thus, the configuration step must be as fast as possible, and it must lead to an efficient combination of complex methods, algorithms and hyper-parameters, to obtain valuable information on the product. The proposed KBS and its inference engine, a bayesian network, is detailed and applied to a product developed by automotive industry. The KBS propose new efficient configurations to achieve designers' goal. This application shorten the configuration step of the NDoE process, and enables designers to use more complex methods. It also allows designers to capitalize knowledge and learn from each past NDoE process.

Keywords: Knowledge based system; Numerical design of experiments; Bayesian network

1. Introduction and context: numerical design of experiments configuration process

Numerical simulation has become more and more used in a mechanical product development process. The product's behavior can be simulated at early stage of this process. This leads to make better and faster decisions about the product design. The product is optimized, its behavior is better understood, and less physical prototypes are required. Recent evolutions in computational capabilities, as High Performance Computing and cloud-computing, boosted the use of complex and accurate numerical models. Multi-physical and multi-levels models simulations of complex product and processes are more and more common, including many components, many interfaces and subjected to coupled phenomena.

Numerical Designs of Experiments (NDoE) consist in applying DoE methodology on a numerical model of the product. NDoE are used to take into account uncertainties and variabilities of product's properties and performances. NDoE are used, for instance, to optimize the product (Hu, Yao and Hua, 2008) or enhance its robustness (Patelli *et al.*, 2012) at an early stage of the design process, without using any physical prototype.

A NDoE is defined by an ordered sequence of simulations from a parameterized numerical model. Each simulation/experiment is defined by a specific set of values of model's parameters. A NDoE process is defined by a NDoE, a numerical model and methods used to analyzed results Plenty of NDoE types exist, as factorial, Box-Behnken, Plackett-Burman or Doehlert designs, but also as space-filling designs, such as Latin hypercubes, low discrepancy sequences or maximal entropy designs (Beal, Claeys-Bruno and Sergent, 2014; Garud, Karimi and Kraft, 2017; Yondo, Andrés and Valero, 2018). Once the type of the NDoE and the number of experiment are defined, simulations are run, and results are obtained. Methods used to analyze these results must be also defined, in accordance with the NDoE (e.g. optimization algorithms, statistical methods, regression methods, etc.).

Despite the usefulness of NDoE process to improve the design of products, two major drawbacks limit its application.

First, a NDoE may be very expensive with complex and accurate simulations. Despite NDoE are used to organize and reduce the number of simulations, hundreds of complex simulations can be easily scheduled and automatically run. Also, uncertainty analysis may require numerous simulations to obtain accurate results. Computing resources are allocated during a long period for each NDoE process. If many long experiments are required, remaining computing resources may be insufficient for other projects.

Second, NDoE increase the amount of generated data drastically. By executing complex and expensive simulations (e.g. a forming process, which can be modeled as a dynamic and non-linear analysis of a multi-components system subjected to heat transfers and plasticity), the company must cope with complex input data and many results. Applying NDoE methodology on such simulations multiplies this first large amount of data by the number of experiments. Data about the NDoE process itself, for process traceability, and outputs produced by analysis of results must also be managed,. All of these data must be managed to be shared and traced during the design process.

The first drawback can be solved with efficient methods of NDoE results analysis, in order to:

- Reduce the number of relevant parameters (factors), and thus, the number of required experiments for future studies. By a sensitivity analysis, designers can focus on the most significant factors (and interactions) (Patelli *et al.*, 2012) (Iooss and Lemaître, 2015).
- Produce a surrogate model (or metamodel) by regression methods. A surrogate model replaces the initial numerical model by a simpler function, faster to be run, such as polynomial function or kriging (Castric *et al.*, 2012) (Yondo, Andrés and Valero, 2018). Once the surrogate model is validated, it is used during future studies to obtain results fast and to shorten decision processes. It is also a way to capitalize the knowledge about the product.
- Define the most useful NDoE. Designers must define the optimal sampling of the space formed by the set of factors. Adaptive NDoE are used to reduce the number of experiments, by defining iteratively optimal experiments (Forrester and Keane, 2009). For instance, a surrogate model can be improved by adaptive NDoE. If the surrogate model, based on a first NDoE, is not accurate enough, a new experiment is selected by an optimization algorithm. The new experiment is added to the initial NDoE, to improve the surrogate model. The selected experiment provides the best improvement for the surrogate model.

These three ways contribute to shorten the NDoE process and decrease its computational cost. But, for each of them, many choices are demanded to the designer with many possibilities. The configuration of the NDoE process must be as efficient as possible. An efficient configuration leads to a NDoE process with minimal cost and relevant results. The configuration covers the selection of the type of NDoE (e.g. factorial design, latin hypercube sampling, space-filling sampling, etc.), the number of experiments, the type of surrogate model (and its specific parameters), the method for sensitivity analysis, the optimization algorithm for adaptive NDoE, etc. (Figure 1). These choices depend on the nature of the studied product (e.g. a complex behavior from a multi-physic model, many factors from a complex product), available computing resources and on the objective of the NDoE process (sensitivity analysis, surrogate modeling, product optimization, robustness analysis, etc.). An ill-configured NDoE process would lead to a loss of computing resources and useless results. All of these choices of methods, which are used to shorten the execution of the NDoE, may extend the configuration step.

The configuration step requires expert knowledge for a fast definition of the best combination of methods which lead to an optimal NDoE process, fast and valuable. A lack of knowledge may limit the use of NDoE processes for industrial uses and complex products and processes.

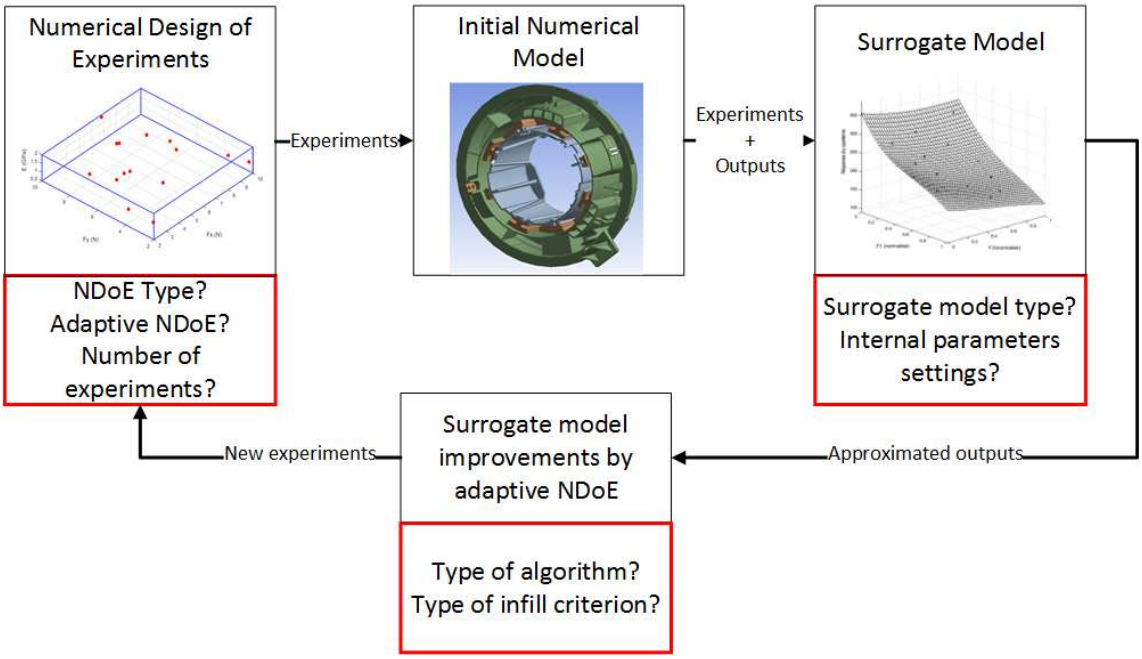


Figure 1: Example of NDoE configuration problem for surrogate modelling.

Some recommendations and guides about this configuration problem exist, but they are not exhaustive (Sanchez and Wan, 2012). Furthermore, some methods are subjected to a random behavior, as metaheuristics used for adaptive NDoE. An optimization approach was considered by (Gorissen, Dhaene and Turck, 2009). The authors focus on surrogate model type selection, based on an evolutionary optimization algorithm and on an adaptive NDoE. This strategy cannot be applied to every element of the configuration. For instance, the selection of the NDoE type by an optimization algorithm would lead to a very expensive process, since each type of NDoE requires a completely new set of experiments which must be executed. Another solution consists in reusing knowledge and data capitalized by the enterprise (Blondet *et al.*, 2015) to increase the profitability of designers' activities.

The knowledge, concerning the NDoE process configuration, can be capitalized and reused by Knowledge-Based Systems (KBS). KBS are a specific branch of Artificial Intelligence (AI) (Kiritsis, 1995), which aims to solve problems faster than humans by exploiting data and knowledge more efficiently. A KBS can analyze knowledge by different reasoning strategies, as symbolic, statistical, connectionist (networks) and distributed intelligence approaches (e.g. multi-agent systems, swarm intelligence, etc.). Specific methods and systems must be used to gather, classify, trace and deliver a large amount of data and knowledge for different stakeholders in extended enterprises, during long periods.

A KBS dedicated for NDoE processes may enhance the profitability of every product design processes in a company:

- NDoE processes may be defined faster, with maximal efficiency.
- More complex and efficient methods may be used easily.
- The product is optimized earlier in the product development process.

- Data and knowledge about the product and NDoE processes are capitalized and can be reused for continuous improvements.
- Human resources and computing resources are used more efficiently.
- Designers can learn from the knowledge base to improve their skills.

This paper presents a proposal of KBS dedicated to NDoE processes to solve the two identified drawbacks which prevent designers from using NDoE processes for complex products. The proposed KBS aims to shorten both the execution step and the configuration step of the NDoE process with data and knowledge management approaches. Section 2 is a literature review of AI methods to select the most appropriate reasoning method to reuse knowledge. Bayesian networks were chosen amongst main types of reasoning methods to be used as an inference engine. The Section 3 details the architecture, behavior and functionalities of the proposed KBS. Section 4 illustrates this proposal by an application on a mechanical product from automotive industries.

2. Design of the knowledge-based system

Knowledge management can be defined as “the creation and subsequent management of an environment which encourages knowledge to be created, shared, learnt, enhanced, organized for the benefit of the organization and its customers”(Sarrafzadeh, Martin and Hazeri, 2006). This is the main goal of the proposed KBS, specifically for NDoE processes. This section focuses on methods to reuse, improve and create new knowledge about NDoE processes.

Knowledge is a major resource for companies. Knowledge management addresses several issues (Dalkir, 2005):

- The extended enterprise relies on collaborative processes, which are based on knowledge sharing with the same comprehension between every stakeholders.
- Knowledge must be capitalized and reused for competitiveness of companies.
- Knowledge must not be lost. A knowledge base can be built to avoid knowledge losses when, for instance, an employee leaves the company.
- Flows of information increases due to continuous progress of computing technologies. Companies must be more responsive to follow this progression.

Knowledge management can be modeled by a three-stage-cycle (Dalkir, 2005) (Figure 2) . Created knowledge is assessed, regarding to its validity and its usefulness. Valid knowledge can be shared across the organization. The contextualization ensures traceability and adaptation to users' needs. Knowledge is reused and user can give a feedback to update and improve the knowledge.

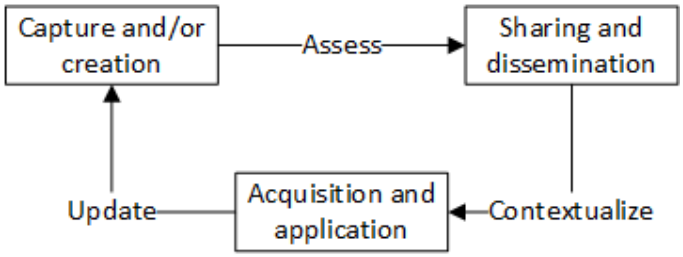


Figure 2: Knowledge management cycle, adapted from (Dalkir, 2005).

Personal knowledge of each co-worker is capitalized, assessed, traced, shared, reused and improved within a global system. Knowledge dissemination and acquisition may demand several transformations. A model of transformation of knowledge was proposed by (Nonaka, Toyama and

Byosière, 2001). This model is based on the difference between explicit knowledge and tacit knowledge. Explicit knowledge can be easily disseminated and reproduced with written documents, manuals, procedures, schematics, etc. Tacit knowledge consists of personal experiences, know-how learned by practice, expertise. Tacit knowledge is harder to be expressed and shared without any loss. Another type of knowledge can be considered, the implicit knowledge, which is defined as the knowledge directly deduced from explicit knowledge, but not explicitly described. The difference between tacit, implicit and explicit knowledge is widely discussed by (Davies, 2015). The cycle of transformation between tacit and explicit knowledge (Figure 3) is built on four stages:

1. Socialization: tacit knowledge is directly taught and learned by dialogue, observation, etc.
2. Externalization: tacit knowledge is translated to explicit knowledge, able to be capitalized by the organization, with, for instance, written documents.
3. Combination: Explicit knowledge is collected and processed to create new explicit knowledge. Gathering data to solve an equation and analyzing results is an example of knowledge combination.
4. Internalization: Explicit knowledge is acquired, learned, experienced and understood by users. This explicit knowledge is transformed into know-how.

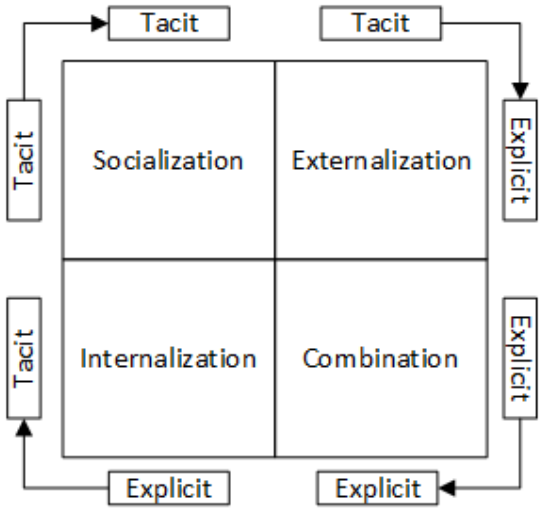


Figure 3: Knowledge transformation cycle, adapted from (Nonaka, 2001).

Designers need advanced knowledge and know-how to define the most efficient NDoE process fast. An automatic process able to combine explicit knowledge and propose solution to designers may help them for this task. The automatic combination of explicit knowledge can be managed by AI methods, such as logical expert systems and machine learning methods. Applications of automatic reasoning methods are more and more numerous in the domain of engineering (Salehi and Burgueño, 2018). While many expert systems based on logical reasoning were developed in the 90's (Wagner, 2017), AI reasoning methods are now massively applied to improve product design processes. These methods are able to learn from gathered data, to manage uncertainty, to recognize patterns and to analyze large amount of data (i.e. with the application of Deep Learning methods).

One method must be selected to define a KBS for NDoE process. In the next sub-section, some of the main AI methods are compared to choose the most appropriate, among logical reasoning, uncertain reasoning and machine learning sub-domains (Figure 4). Several sub-domains were ignored, such as:

- Optimization algorithms. They are not adapted for the NDoE process configuration, as discussed in the first section;
- Text and image recognition, and robotics, which are not relevant in our context;

- Deep learning methods. Data on engineering practices may be too rare in industries to deploy deep learning methods.

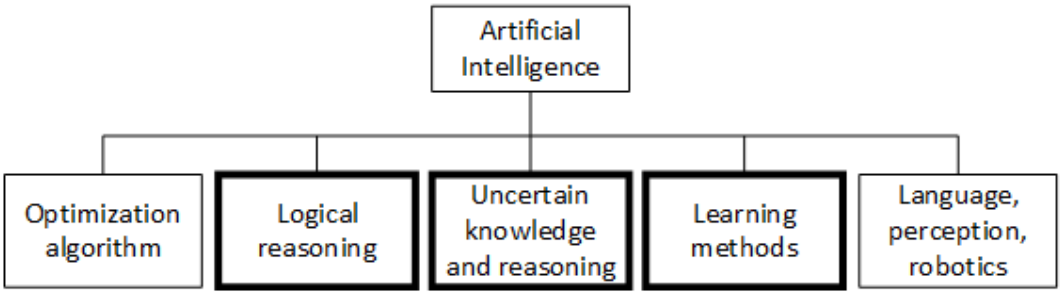


Figure 4: Main areas in Artificial Intelligence (Russell, 2010).

2.1 Comparison and selection

In this sub-section, main types of reasoning methods are compared according to some criteria to choose the most appropriate method to configure the NDoE process. The comparison follows the classification of AI domains, developed in (Russell, 2010) and focuses on logic reasoning methods, methods able to manage uncertain knowledge and machine learning methods. Logical reasoning, neural networks, decision trees, bayesian networks and case-based reasoning are compared according to 5 criteria. These criteria were defined in relation with partners in automotive and energy industries. These criteria must be fulfilled to ensure that the proposed KBS is adapted to help designers for NDoE configuration step:

1. **Explicit inference**. The KBS is must not replace the designer. It must provide help and advices. It must be able to explain clearly its reasoning.

2. **Integrated Expert knowledge**. Existing expert knowledge in a company must be integrated in the KBS as well as possible. At least, explicit knowledge should be supported by the system. Nevertheless, the existence of expert knowledge should not be mandatory for the KBS Unknown methods must not be disqualified because they were never used nor studied.. In this way, exploring new methods could be incited.

3. **Learning**. The proposed KBS must support the lack of expert knowledge. It must be able to learn from previous executed NDoE processes. This empirical type of reasoning would compensate the lack of prior knowledge, let the system propose innovative NDoE configurations and let it discover new knowledge.

4. **Discrete and continuous data support**. Most of choices designers must make are discrete. For instance, the type of NDoE is limited to categorical choices: full factorial, Box-Behnken, Latin Hypercube Sampling, etc. Some others variables are continuous, as convergence thresholds for optimization algorithms, or performance indicators of the NDoE process, as execution time or accuracy and predictivity of a surrogate model.

5. **Random behavior support**. Some algorithms used during the NDoE process have a random behavior, as metaheuristics used for adaptive NDoE (evolutionary or swarm-based algorithms). Thus, identical NDoE configurations may lead to different, or contradictory, results and performances. The KBS should not be too sensitive to this phenomenon.

2.1.1 Logical-rule based expert systems

Several expert systems based on formal logic reasoning were developed to provide a decision-aid system for design of experiments. DEXPERT (Lorenzen *et al.*, 1992) is an expert system based on a set of logical rules. It assists designers in defining physical experiments. It covers DoE objectives as sensitivity and robustness analysis, optimization and surrogate modeling, but methods supported to fulfill these objectives are limited (e.g. few DoE types are considered, the only type of available surrogate model is polynomial model, etc.). DEXPERT embeds an interface, customized according to the user's profile, which propose an efficient DoE configuration and explain the reasons of this proposition. Every change of the DoE are capitalized and traced, but these data are not reused to improve the system. DEXPERT is unable to learn from capitalized data.

Similar systems were proposed by (Chen, 1991; Weiner, 1992; Naranje and Kumar, 2014) and have the same limitations. The main limitation of this approach is the requirement of an exhaustive and accurate definition of rules. A second limitation is its inflexible behavior, which can dismiss several solutions which may be good. Fuzzy logic alleviates these limitations by taking into account the inaccuracy of the knowledge(Urrea, Henríquez and Jamett, 2015)proposed a fuzzy expert system to select automatically the most adapted materials for a mechanical structure. The integration of fuzzy logic is an evolution for expert systems, useful to represent uncertain rules.

More recently, a learning ability was added to rule-based expert systems, by using the Inductive Logic Programming (ILP) method. Such a system was used for the numerical simulation process (Dolšak, 2002). FEMDES (Finite Element Mesh Design Expert System) assists the designer during the meshing step to shorten the simulation process. The rule base embeds 1900 rules induced by ILP method and ten reference meshes. Thus, logic rule-based systems are able to learn from data, but it required coherent and validated data to build rules. Since some algorithms used in NDoE processes may have a random behavior, rule-based systems may be too sensitive to be applied.

To conclude, logical expert systems do not fulfill criteria n°2, since expert knowledge is mandatory and not optional in expert systems, n°4, since expert systems cannot handle continuous data, and n°5 because of their sensitivity to randomness.

2.1.2 Artificial neural networks

Artificial Neural Networks (ANN) were inspired by brain's neurons. An artificial neuron receives inputs from several sources and transform the combination of these inputs if this combination exceeds a given threshold. The principle of ANN is to link numerous neurons to each other, with a specific structure defined according to the type of problem to be solved. Many types of ANN exists (Schmidhuber, 2015; Tkáč and Verner, 2016), such as multi-layer feedforward networks, recurrent ANN, convolutional ANN, radial basis functions networks, generative adversarial networks, and many others. All of these types have different architectures and learning strategies to achieve different objectives, as regression, classification, pattern recognition (image, text, event), for different purposes, as detection, prediction and recommendation.

ANN were applied to determine the best surrogate modeling method according to a specific problem (Cui *et al.*, 2016). Deep learning method was used to improve finite-element method in computational mechanics (Oishi and Yagawa, 2017). ANN were also used to predict gasoline engine performance (Tasdemir *et al.*, 2011). Another application for manufacturing uses this method for feature recognition and process planning (Ding and Matthews, 2009).

ANN are able to automatically learn from data, even from noisy and incomplete data. ANN can handle complex behavior with multiple variables. But, the main drawback of ANN is their "black-box"

aspect. It is very difficult to explain a decision made from an ANN. Thus, the criterion n°1 is not fulfilled by ANN. It is also difficult to explicitly add expert knowledge inside the ANN.

2.1.3 Decision trees

Decision trees are based on a hierarchical sequence of tests performs on learning data, to build a tree. Different types of decision tree algorithms exist, as CART (Breiman, 2001), C4.5 (Quinlan, 1996), CHAID (Kass, 1980), and also several evolutions, as random forest. Decision trees can be used for feature selection and classification, for instance, for fault diagnosis (Sakthivel, Sugumaran and Babudevasenapati, 2010).

These methods are non-parametric (no hypothesis on data distribution), non-linear, robust and easy to understand by a clear graphical representation. Over-fitting effects can be limited by corrective methods, as pruning methods (Sahin, Tolun and Hassanpour, 2012). But, it is impossible to pre-define the structure of the tree. The structure is computed automatically. Thus, a decision tree cannot be enriched by user's knowledge (Bayat *et al.*, 2009). The criterion n°2 is not fulfilled.

2.1.4 Bayesian network

Bayesian network is defined as a directed acyclic graph. This graph is composed of nodes (linked to a probability distribution) which are connected to each other to illustrate dependencies and causations (linked to conditional probability distributions) (Naïm *et al.*, 2007; Russell, 2010). A bayesian network is built with a two-steps approach. First, the structure (the graph) is determined. It can be determined with manual settings (a given relationship is forbidden, or demanded) done by an expert, and learned from data. Concerning structure learning, many algorithms exists, mainly based on statistical independence tests or on a score (Naïm *et al.*, 2007; Liu *et al.*, 2017). Second, probability distributions are determined according to the structure. These distributions can be based on expert knowledge and/or computed by statistical methods (e.g. maximum likelihood estimator).

Bayesian networks were used for process optimization. For instance, they were used in an adaptive DoE process to define new experiment for high-dimensional problems (Slanzi and Poli, 2014). Bayesian networks are also more and more used for manufacturing processes (Poeschl *et al.*, 2017), to improve machine assignments (Hanafy and ElMaraghy, 2014) or to predict the quality of a machining process (Correa, Bielza and Pamies-Teixeira, 2009).

Bayesian networks can represent uncertain knowledge with clarity by a graph and probability tables. The structure of the network and probability distributions can be defined by the user according to its knowledge. Then, the structure and distributions are completed by learning algorithms. Bayesian networks are able to determine as discrete as continuous variables. Bayesian networks fulfil every criterion.

2.1.5 Case-based reasoning

Case-based reasoning is a method to reuse and adapt solutions applied to former problems for a new problem. This type of method is composed of 4 main steps (Aamodt and Plaza, 1994):

1. Retrieve similar cases. Many methods to measure the similarity exists (Lopez de Mantaras *et al.*, 2005);
2. Reuse knowledge linked to these similar cases to solve the new problem. It can be done by reusing the solution used for the most similar case, or by reusing the method which generated the solution for the most similar case;

3. Revise the solution or the method to fit to the new problem;
4. Retain and memorize this new case to enrich the knowledge base.

This type of reasoning method was applied for numerical simulation, to propose automatically a mesh for finite-element models (Khan, Chaudhry and Sarosh, 2014), or to improve efficiency of mechanical design processes

Case-based reasoning methods are able to learn from data and to give clear explanation of proposed solutions. However, expert knowledge is mandatory to define rules to reuse and adapt a solution for a new problem. Expert knowledge is not optional, and the criterion n°2 is not fulfilled.

2.1.6 Hybrid systems

Hybrid KBS combine several approaches in their inference engine. A large amount of different hybridization has been developed in the last decade (Sahin, Tolun and Hassanpour, 2012; Tkáč and Verner, 2016). The aim of this approach is to propose a more efficient inference engine, which combine advantages of many different systems. They can combine, for example, ANN, decision trees, formal logic, fuzzy logic or evolutionary algorithms. Hybrid systems can have various properties, following integrated methods. They were not included into the comparison for this first attempt.

2.2 Reasoning method selection

This review covers main methods for data and knowledge analysis, in order to select one of them and to use it as an inference engine. It is a first attempt to define a relevant reasoning strategy. At this step, the covered methods were not compared in terms of predictive performance.

Bayesian networks satisfy every criterion previously defined (Table 1). They are able to give clear explanation of decision; they can learn from data, they can be enriched by expert knowledge if it exists in the company; they can represent uncertainties, for instance, about the efficiency of a NDoE configuration with random behavior, and they support both continuous and discrete data.

Table 1: Comparison between knowledge analysis methods.

Type of reasoning	Criterion / Method	N°1	N°2	N°3	N°4	N°5
Logic-based expert systems	Formal-logic, fuzzy logic, ILP	✔		✔		
Machine learning	Artificial Neural Networks			✔	✔	✔
	Decision trees	✔		✔	✔	✔
	Bayesian networks	✔	✔	✔	✔	✔
Case-based reasoning		✔		✔	✔	✔

Criteria: 1- Explicit inference; 2-Optional support of expert knowledge; 3- Learning; 4- Discrete and continuous data support; 5- random behavior support.

This review can be completed by studying the large variety of other approaches and complex strategies of reasoning. Hybrid systems have been temporarily dismissed, and many other algorithms exists in machine learning (non-parametric models, support vector machine, etc.). Other strategies, such as unsupervised learning and reinforcement learning methods could be also useful. Unsupervised learning can classify data without any results (before the execution of an NDoE configuration). Reinforcement learning methods may improve the learning phase by a strategy of

reward and punishment, which may be more responsive than supervised learning to learn on completely new configuration.

The next section details the application of bayesian networks to predict an efficient configuration of NDoE process and to assess the efficiency of a specific configuration, to help designers to set and execute this process faster.

3 A bayesian KBS for NDoE processes

3.1 Global architecture

The general architecture of the proposed KBS is composed of the NDoE process, a knowledge base and inference engine (Figure 5).

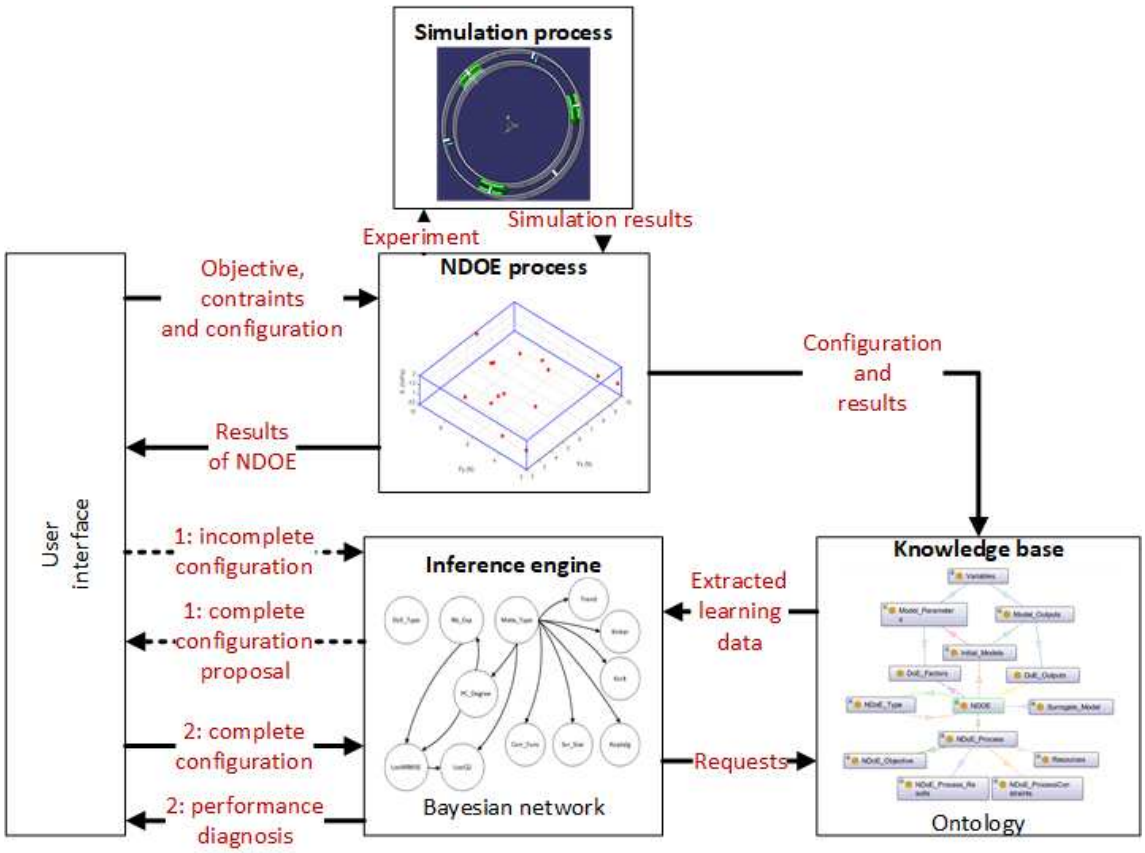


Figure 5: Architecture of the proposed KBS for NDoE processes.

The NDoE process is based on a validated parameterized numerical model and a simulation process able to run each experiment. The process must be completely defined to be executed. The definition covers an objective (e.g. sensitivity analysis, surrogate modeling, optimization, etc.), constraints (e.g. time limit, a targeted accuracy of a surrogate model, etc.) and the configuration of the methods involved in the NDoE process. The configuration consists in defining methods and algorithms required to fulfill the objective and constraints. For instance, a NDoE process configuration may be defined by a number of experiments, a specific sampling method, a surrogate model type, etc. Once the NDoE process is defined, experiments are executed and outputs are computed. These data (i.e.

each experiment associated with its results) are analyzed to obtain final results (e.g. a surrogate model, or an optimal product design).

A user interface, not represented here, is used to ensure a bi-directional communication with the user. The user is able to define a known part of the NDoE process, to ask the system for a complete configuration with explanations, and to validate or modify the proposed configuration.

The knowledge base is supported by a specific ontology for NDoE, which was developed in a previous work to support knowledge externalization and internalization steps (Blondet *et al.*, 2018). The inference engine reasons on the knowledge to assist the designer to set the most efficient NDoE process according to a specific problem. As discussed in the previous section, bayesian networks are the most appropriate method for this application.

The ontology and the bayesian inference engine are described further.

3.2 Ontology

Amongst existing definitions of ontology, the term can be defined as a description of a domain of knowledge used by a community of agent, with proper concepts and relation between these concepts (El Kadiri and Kiritsis, 2015). In our context, concepts are every type of data and metadata related to NDoE processes, and agents mean engineers, designers and, more generally, users of NDoE.

By means of this ontology, knowledge and data are gathered, structured, traced and shared in a comprehensive way in collaborative organization. Every NDoE process is an instance of this ontology. The design of this ontology was motivated by 3 reasons:

1. It provides a first logical reasoning step. The ontology is based on OWL (Web Ontology Language), which enables the use of logical reasonner to check if the knowledge base is coherent, and to make logical deductions on concepts and instances.
2. It supports knowledge sharing between users by a common comprehension of concepts. Clear semantic relationships between concepts can be defined to enhance knowledge sharing between different teams and departments involved in the development of a specific product.
3. The description of concepts does not depend on the technical implementation and can be reused in other contexts. A common core of concepts and semantic relationships is defined and can be specialized in different contexts.

The global view of the ontology (Figure 6) shows the main concepts of the semantic domain of NDoE. A specific taxonomy was developed and enriched by semantic relationships. For instance, the class "NDoE Type" is linked to its own sub-classes, which are the different types of NDoE (e.g. full factorial design, Latin Hypercube Sampling, etc.). More complex semantic relationships are added to this taxonomy. For instance, a NDoE is defined by factors, a NDoE type, and can be used to compute a surrogate model. All of these relationships are used to check the consistency and the coherence of the ontology and its instances. Each instance is a NDoE process, with its configuration, its objective, its constraints and its results.

An ontology is also supposed to be linked to other existing ontologies. An ontology must be consistent with others ontologies to form an extended semantic description of a wider domain. The ontology developed for the KBS is linked to other ontologies with appropriate semantic relationships. STATO[1] and EXPO (Soldatova and King, 2006; Vanschoren and Soldatova, 2010) detail DoE, statistical methods and algorithms. The process-oriented ontology PARO (Le Duigou and Bernard, 2011) is also linked to the proposed ontology. For instance, PARO provides a detailed description of the concept "Resources", as human and technical resources available in a project.

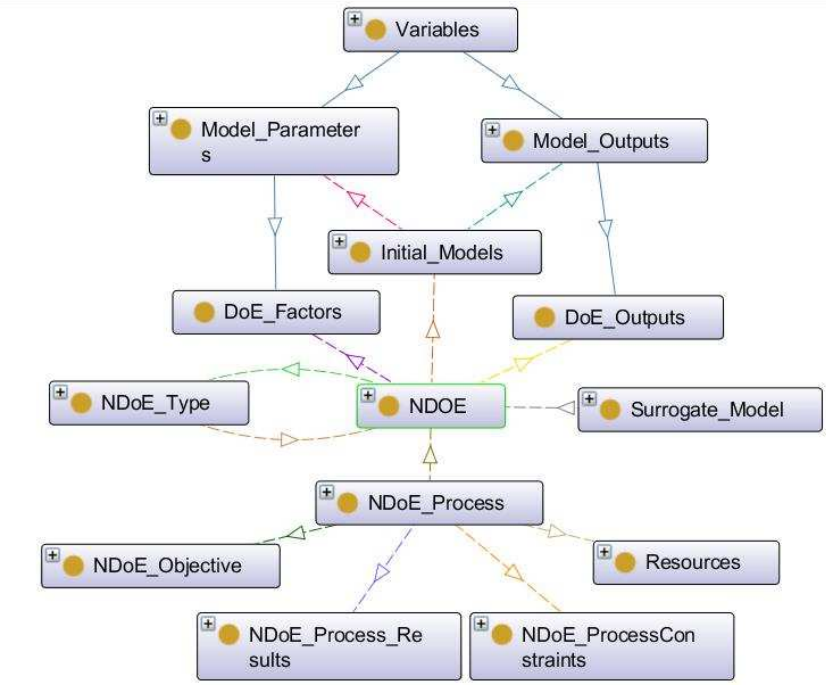


Figure 6: Global view of the ontology developed for the KBS (arrows are semantic relationships).

The knowledge base contains all data required to define the NDoE process, in connection with product and project data. More specifically, it contains the entire definition of each NDoE process, results from experiments and analysis, and performance indicators (e.g. execution time, accuracy and predictivity measures for a surrogate model, number of added experiments in case of an adaptive NDoE, etc.).

The inference engine, detailed in the next sub-section, is based on a dataset extracted from the ontology.

3.3 Inference engine

The inference engine must be defined to manage the configuration step of the NDoE process. Two modes are considered (Figure 5). First, designers have an incomplete knowledge about the right configuration for their specific context. They may know, for example, what kind of surrogate model should be used, but they do not know how many experiments are required. Elements which are missing in the configuration are proposed by the inference engine, according to the context (e.g. type

[1] https://www.ebi.ac.uk/ols/ontologies/stato

of product), the objective (e.g. to produce a surrogate model) and some constraints (e.g. limited resources, time limit, accuracy required for results). In the second mode, the inference engine starts with a complete user-defined configuration. The inference engine predicts the performance of this configuration, such as execution time, the accuracy of the results, etc. If these performance indicators do not match with designers' requirements, the KBS switch on the first mode to propose new and better configurations. These predictions must be proposed with some indicators to inform the designer whether these predictions are trustworthy or not.

In order to set a bayesian network to assist designers in defining a NDoE process, three steps are required: (1) the selection and the definition of the network's variables; (2) the estimation of the structure and probability distributions by learning from data and/or adding expert knowledge; and (3) the validation of the bayesian network, to control the trustworthiness of its predictions. Next sub-sections give details about these three steps. More generally, the process to define the bayesian network follows Knowledge Discovery in Databases (KDD) methodology (Lara *et al.*, 2014). Available data extracted in the knowledge base must be filtered, cleaned and transformed to be analyzed efficiently.

3.3.1 Definition of bayesian network variables

As a first step, variables must be defined. In this context, each variable is an element of the configuration of the NDoE process, or a measurement of its performance (e.g. execution time, accuracy of a surrogate model, etc.).

These variables are selected according to designers' needs. If they do not know what type of DoE should be selected, the variable "DoE_Type" must be included. Then, it depends on the objective of the NDoE process. For instance, if the objective is to produce a surrogate model, a variable describing the type of sensitivity analysis is useless. The main goal of variable selection is to minimize the number of variables to keep the network as simple as possible.

Three types of bayesian network's variables are considered:

- A discrete qualitative variable has a finite number of possibilities – or events. For example, the variable "DoE_Type" can be defined with a finite number of type of DoE, such as Latin Hypercube Sampling, Full Factorial design, etc.
- A discrete quantitative variable may have a high number of event, like the variable "Number of Experiments" (positive integer). Usually, a NDoE may involve hundreds simulations. If the scope of NDoE methods is extended to Monte Carlo sampling method, the number of simulation could be higher than 10 000 simulations. In this case, an efficient approach is to discretize this set into several smaller sets. The number of sets and their ranges must be carefully chosen, as they can have a significant effect on predictions of the KBS.
- A continuous variable can be modeled by a continuous probability distribution, but it can be also discretized to simplify the reasoning process.

An approach to simplify the bayesian network is to apply a "multi-net" strategy (Naïm *et al.*, 2007). The exhaustive network is cut with the externalization of specific variables. For NDoE process, the variable "NDoE objective" is a discrete qualitative variable. Each event of this variable, such as "surrogate modeling" or "sensitivity analysis" will cause the exclusive use of other variables. Two different, and simpler, sub-networks can be defined for each of these two events.

3.3.2 Learning step

The second step is the determination of the structure and probability distributions to define the bayesian network.

The structure is the set of dependence links between each node (variable) of the network. These links must be directed from one node to another, and they must not create any cycle in the network. The aim of the structure is to model causal relationships between nodes. The structure is determined by expert knowledge and completed by a learning algorithm. The insertion of expert knowledge consists in declaring, a priori, dependence relationships between each couple of variables. A learning algorithm is then used to complete the structure of the network from observations. Many algorithms exist to detect possible links, based on statistical independence test (Spirtes, Glymour and Scheines, 1993; Pearl, 2000), on a score (Akaike, 1970; Schwarz, 1978) or on hybrid algorithms (Scutari, 2014; Madsen et al., 2015). In this paper, the hill-climbing algorithm was considered to determine the structure for the use-case, as a proof of concept. The hill-climbing algorithm is based on a greedy search on possible structures. The selected structures maximize of the Bayesian Information Criterion (BIC) (Scutari, 2010).

Probability distributions may be also estimated according to experts' knowledge, by modeling the real distribution, such as normal or uniform distribution, or by setting manually the probability of each event. Then, missing distributions are estimated from the available data. Different methods exist to estimate these probability distributions, such as maximum likelihood, maximum a posteriori or expectation maximization.

Bayesian estimators, such as the maximum a posteriori estimator, combine a frequentist approach and a probability distribution defined a priori. If a little amount of learning data are available, the probability distribution, defined by an expert or let as a uniform distribution (if the distribution is unknown), become more significant than data. Thus, if an unknown situation occurred, with a completely new context with no corresponding NDoE process, there will never be any impossible configuration. The KBS can be used even if there is a lack of data. This type of estimator was selected for the use-case of this paper for this reason.

3.3.3 Validation step

The validation of the bayesian network consists in a cross-validation to assess its ability to propose good predictions. A common way is to split the dataset in two subsets. The first subset (e.g. 80% of the whole dataset) is used for the learning step, and the second subset is used to validate the learning step (Powers, 2011).

Once the structure and probabilities are completely determined, the bayesian network is assessed on the validation subset, and a series of confusion matrices is computed. Each confusion matrix is specific to a given variable.

Common indicators used in machine learning are computed from this matrix (detailed in (Powers, 2011)), such as the precision (Predicted Positive Value, PPV), the sensitivity (True Positive Rate, TPR), the fall-out (False Positive Rate, FPR), the Negative Predicted Value (NPV) and the Specificity (True Negative Rate, TNR) of the bayesian network, for this variable. These indicators are illustrated by a histogram (Figure 8) to clearly warn designers that the bayesian network, for the selected variable, could give right or wrong predictions.

Some other global classification errors are computed, as a k-fold-out error and a global accuracy. The statistics concerning the data distribution for each variable and event is also shown to check if the dataset is balanced or not.

The next section describes an industrial application of this KBS, in an automotive industry.

4 Use-case

This section shows an application of the proposed KBS on a blower of HVAC system (Heating, Ventilation and Air-Conditioning) developed by an automotive supplier (Figure 7). The finite element model simulates the dynamical behavior of the blower, especially to design elastomer dampers between the rotor and the stator. The goal is to minimize the transfer of vibrations due to dynamic unbalances phenomena.

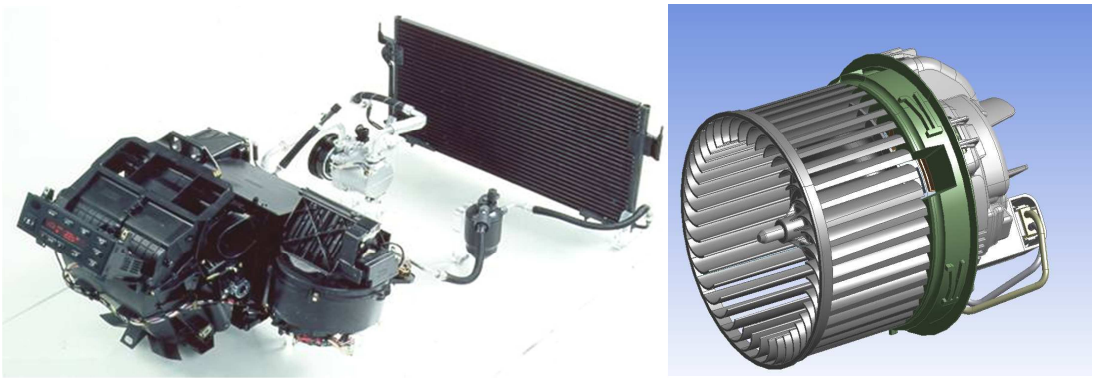


Figure 7: HVAC system (left) and 3D-model of the air-blower (right) (Valeo©).

The company applied NDoE processes to optimize their products, to identify the most influent geometric and material properties on the dynamic behavior, and to produce a surrogate model of the finite element model to accelerate future decision processes. For this finite element model, eight physical parameters, named factors, were defined. These factors covers six geometric parameters, the stiffness of the elastomer and the phase difference between rotor and stator.

The goal of this case-study is to illustrate the usefulness of the application of a KBS based on bayesian networks in the context of the product design process. In this context, few data are available. Thus, the case-study is based on a relatively small dataset. The use-case shows how this KBS could assist designers to configure their NDoE process faster to produce an effective surrogate model of the finite-element model of the air-blower. Then, the surrogate model is used interactively to make decision fast during meetings. Thanks to the surrogate model, the behavior of the product is stored in a lighter and more neutral form. There is no need to use the simulation software, or to keep it operational, to regenerate the results after several decades.

The proposed KBS was implemented with following components:

- The inference engine, a bayesian network, is based on the R package 'bnlearn' (Scutari, 2010; Denis and Scutari, 2014);
- The knowledge base was designed in Protégé 5 and used with Virtuoso Universal Server®;
- The NDoE process was built on a set of Python scripts in connection with the software Uranie (Gaudier, 2010). Each experiment/simulation is done by the finite element method solver Code_Aster[2].

4.1 Definition of variables and generation of the dataset

[2] http ://www.code-aster.org/

The variables of NDoE process configuration were defined according to the surrogate modeling objective and the available methods in Uranie (Table 2). NDoE process configurations were generated randomly according to variables and event defined in Table 2. For instance, a configuration can be composed of a NDoE based on Latin Hypercube Sampling, with 17 experiments, a polynomial chaos model with a maximal degree of 2.

Table 2: Variables of NDoE process configuration considered in the use-case and associated events.

Name	Events	Description	Configuration generation
Type of DoE	Latin Hypercube Sampling ; Halton sequence ; Sobol sequence.	Type of sampling method to define each experiment.	Equiprobable choice.
Number of experiments	[5-15] ; [16-26] ; [27-37] ; [38-50]	Size of the NDoE in terms of scheduled simulations.	Equiprobable choice in [5-50].
Type of surrogate model	Polynomial chaos ; Kriging	Method to compute a regression model on experiments' results.	Equiprobable choice.
Polynomial chaos degree	0 (no) ; 1 ; 2	The maximal degree of the polynomial basis. The degree 0 means the kriging is used.	Depending on the number of experiments. <25 : degree = 1. >25: degree = 2 .
Kriging_Trend (trend function)	0 (no) ; constant ; linear	The hyper parameters of kriging surrogate model implemented in Uranie. The optimization algorithm is used to compute the surrogate model from experiments. 0 means the polynomial chaos model was used.	Equiprobable choice.
Kriging Corr_ Func (Correlation function)	0 (no) ; Gauss ; exponential ; Matern 3/2 ; Matern 5/2 ; Matern 7/2		Equiprobable choice.
Kriging kniter (number of iteration)	0 (no) ; [1;500] ; [501;1000] ; [1001;1500]		Equiprobable choice between 100, 200, 500, 800, 1000, 1200, 1500
Kriging criterion	0 (no) ; LOO ; ML ; ReML		Equiprobable choice.
Kriging Optimization algorithm	0 (no) ; BFGS ; NelderMead ; BOBYQA		Equiprobable choice.
Kriging Scr_size (Screening size)	0 (no) ; [1;250] ; [251;500]		Equiprobable choice between 100, 200, 300, 400 and 500
Output variables			
Surrogate model accuracy (LooNRMSE)	<0.001 ; [0.001,0.01] ; [0.01;0.05] ; >0.05 ; FAIL	This estimates accuracy of the surrogate model. It is obtained by Leave-One-Out (Loo) Normalized Root Mean Squared Error. The FAIL event means an error occurred.	
Surrogate model predictivity	<0.90 ; 0.90-0.99 ; >0.99 ; FAIL	This estimates the predictivity of the surrogate model. It is obtained by Leave-One-Out (Loo)	

(LooQ²)		with the Q² estimator (Iooss and Lemaître, 2015). The FAIL event means an error occurred. The closest to 1 is the best.	

Based on air-blower finite-element model, 2000 NDoE processes were executed to create a first dataset to assess our proposal. 2000 configurations is a high number since the capitalization of NDoE data is rare in mechanical industries, but it covers almost 0.5% of possible configurations only, in accordance with considered variables and events. The aim of this KBS is to help designers even with a small dataset.

Each configuration had the same probability to be generated. The only exception concerns the degree of polynomial chaos models. The degree and the number of experiment are dependent. The number of experiments n, required to compute coefficients of the polynomial models, is determined by $n = \frac{(d+p)!}{d!p!}$, where p is the degree of the polynomial basis, and q is the number of factors of the finite element model. Thus, with 8 factors, 45 experiments are required for a degree of 2. In this use-case, mistakes were deliberately added in the dataset to simulate some human errors and to assess the ability of the KBS to discover the rule by itself. This equation was not followed to generate the dataset. But, to generate the dataset, the degree of the polynomial basis can be 2 from 25 experiments.

This dataset does not aim to be complete and perfect. Such a dataset may be considered as a large dataset in mechanical design departments. Thus, even with uniform distributions to generate NDoE process configurations, some methods actually occurs more often than others and some specific configurations were not in the dataset. Some variables of the bayesian networks may suffer from lack of data. These imperfections were deliberately kept to illustrate more realistic situations. These NDoE configurations were executed, and stored and structured with their results in the ontology (Blondet *et al.*, 2018).

Relevant data are extracted from the ontology with SPARQL[3] requests. These data are selected regarding the context of the current study (e.g. type of simulation model) and the objective of the NDoE process. It concerns, for this case study, every NDoE process based on models of dynamic behavior used to produce a surrogate model.

4.2 Learning step

The learning dataset is 80% of the full dataset. The structure is determined, in this case, only by learning from data. No expert knowledge was included. The hill-climbing algorithm was used to estimate the graph, as shown in Figure 8. This graph shows the dependencies between the variables of the configuration of the NDoE process. The full dataset is not supposed to be complete nor balanced. This choice aims to reproduce a realistic situation during a product design process. Thus, the graph of the bayesian network is an estimation at a given time. This graph will be improved with the progressive enrichment of the knowledge base.

[3] SPARQL stands for SPARQL Protocol and RDF Query Language. It is a recursive acronym. RDF stands for Resource Description Framework.

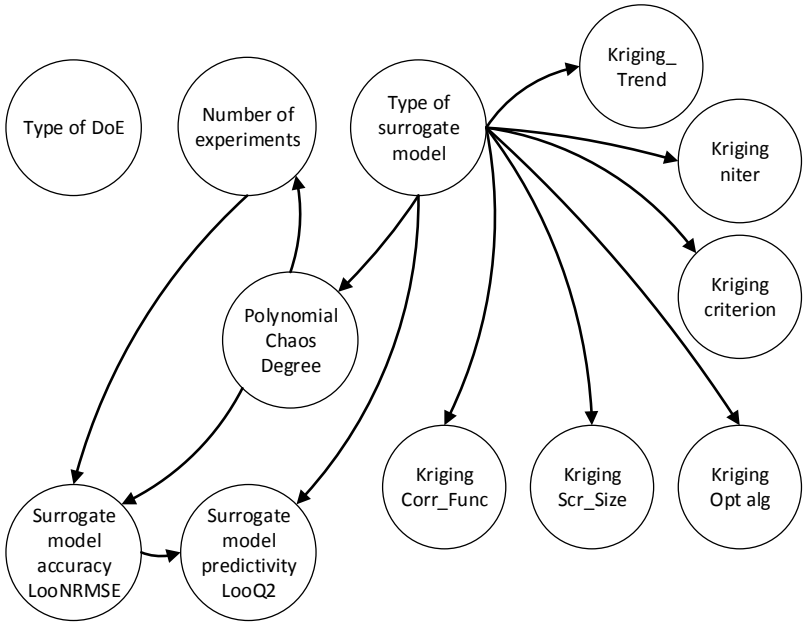


Figure 8: Structure of the bayesian network obtained by Hill-Climbing algorithm.

In this case, the type of DoE has no influence on the quality of the surrogate model. The only expert rule included in the generation of the dataset, which concerns the degree of polynomial chaos model according to the number of experiment, has been recognized by the hill-climbing algorithm.

Probability distributions are determined by a bayesian estimator included inside bnlearn packages (Scutari, 2010), combining an estimation from data and from a priori uniform probability distribution. In this use case, the estimator was set to favor data. Events absent from the learning dataset are modeled by uniform distribution, so that the KBS can propose innovative configurations.

4.3 Predictivity of the bayesian network

To inform the designer about the trustworthiness of KBS's predictions, a confusion matrix and performance indicators are computed for each variable/node. Table 3 details these computations for a specific node of the graph.

For this variable, the predictivity of the bayesian network is good, but it depends on the event. First, dataset is unbalanced. Some events have a higher amount of data than the others. Second, the validation dataset is nearly as unbalanced as the learning dataset: these two datasets are consistent. Third, values of performance indicators vary over events. For instance, there will be much more false positive prediction for the event ">0.99" than for the event "<0.90". Moreover, designers are informed that the event "FAIL" shows a risk of overfitting by predicting perfectly the result.

With these indicators, designers have a tool to assess predictions proposed by the bayesian network. They can evaluate the risk caused by making the decision, for example, to follow advice given by the inference engine.

Table 3: Example of confusion matrix of the bayesian network applied for the use-case and performance indicators.

Surrogate model predictivity $LooQ^2$		Predicted by the bayesian network			
	Events	<0.90	0.90-0.99	>0.99	FAIL
Observed in the validation subset	<0.90	31	0	1	0
	0.90-0.99	7	94	24	0
	>0.99	0	6	165	0
	FAIL	0	0	0	72
Dataset					
Number of learning data	1600				
Number of validation data	400				
Indicators for each event					
Data ratio for the learning dataset		6.63%	28.38%	42.94%	22.06%
Data ratio for the validation dataset		8.00%	31.25%	42.75%	18.00%
True Positive		31	94	165	72
False Positive		7	6	25	0
True Negative		361	269	204	328
False Negative		1	21	6	0
TPR: True Positive Rate (sensitivity)		96.88%	75.20%	96.49%	100%
TNR: True Negative Rate (specificity)		98.10%	97.82%	89.08%	100%
FPR: False Positive Rate		1.90%	2.18%	10.92%	0.00%
PPV: Positive Predicted Value (precision)		81.58%	94.00%	86.84%	100%
NPV: Negative Predicted Value		99.72%	89.67%	97.14%	100%
Global indicators					
Global Accuracy		90.5%			
classification error (k-fold-out, k=5, 100 repetitions)		Mean : 11.27%, standard deviation : 0.0006			

Then, a graphical representation is generated and shown to designers (Figure 8). This histogram was generated for the node "Surrogate model predictivity LooQ2. Four events were considered: $Q^2<0.90$ (bad surrogate model), $Q^2 \in [0.90\,;\,0.99]$ (good surrogate model), $Q^2>0.99$ (excellent surrogate model) and “FAIL”, which indicate an error in the surrogate modelling process.

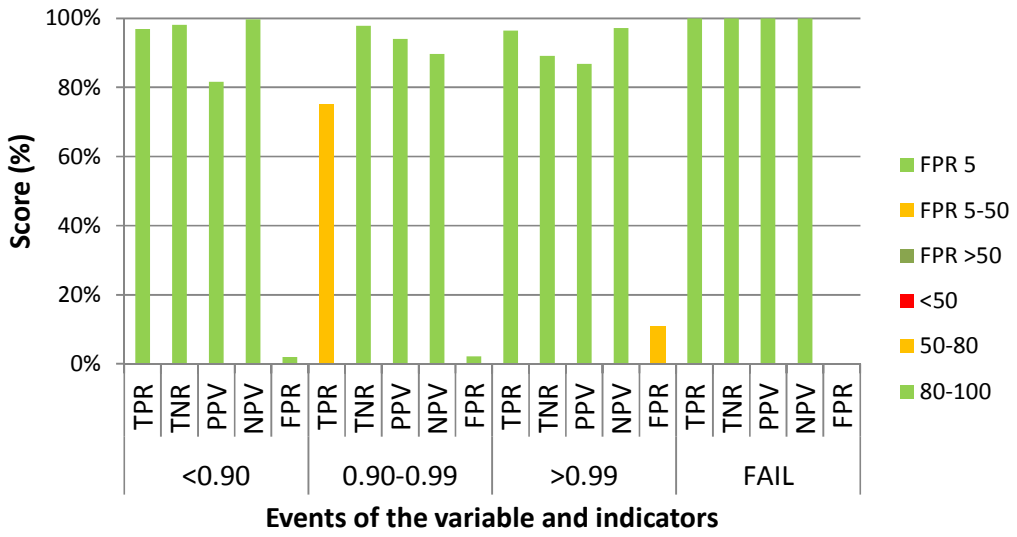


Figure 9: Indicators of the predictive performance of the bayesian network for the variable "surrogate model predictivity".

4.4 Requests to the bayesian network

Designers want to apply a NDoE process on the finite-element model of the air blower to obtain a surrogate model. First, they propose a complete user-defined configuration to predict its performance.

The bayesian network evaluates the probability to obtain the required accuracy (LooNRMSE) and predictivity ($LooQ^2$) level for the surrogate model with the user-defined configuration (Table 4).

Table 4: User-defined configuration proposed to the bayesian network.

Variable of the configuration	Value
Objective	Surrogate modelling
Minimal accuracy (LooNRMSE)	0.05
Minimal predictivity level ($LooQ^2$)	0.98
Strategy	Non-adaptive
Numerical model	Rotor1a
Type of NDoE	Halton
Number of experiments	15
Type of surrogate model	Kriging
Kriging model's internal parameters	
Correlation function	Matern 7/2
Trend function	Linear
Optimization criterion	Leave-One-Out
Optimization algorithm	BOBYQA
Maximal number of iterations	800
Initial screening sampling	200

The probability of this event (Table 5), estimated by bnlearn, is 0 according to this bayesian network. This result is due to an approximated inference algorithm used in bnlearn, based on 100000

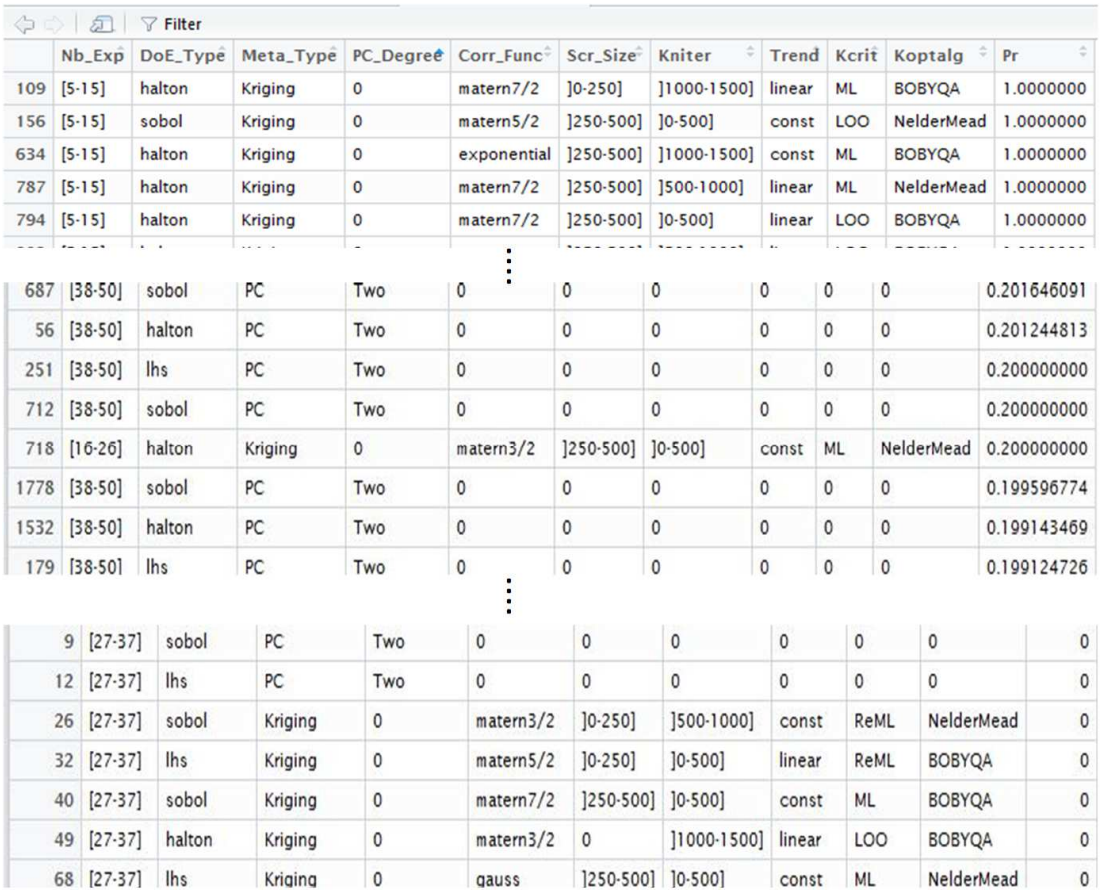
simulated NDoE configurations, obtained by re-sampling techniques. However, it shows it is nearly impossible to obtain required performances with this configuration.

Table 5: Request to predict the performance of the user-defined configuration.

```
cpquery(fitted,
        event = (LooNRMSE == "0.01-0.05" & LooQ2 == "0.90-0.99"),
        evidence= (Nb_Exp == '[5-15]' & DoE_Type =="halton"
        & Meta_Type == "Kriging" & Corr_Func == "matern7/2"
        & Scr_Size == "]0;250]" & Kniter=="]500-1000]"
        & Trend=="linear" & Kcrit =="LOO" & Koptalg =="BOBYQA"), n=100000)
```

Since this configuration does not fulfill designer's requirements, the KBS switch on the first inference mode, to propose new configurations. The use of the bayesian network is reversed to predict a list of the best configurations, by assessing every possible configuration (Figure 10). Designers can focus their choices on about only 10 configurations among thousands possible configurations.

These results shows a low probability of success for polynomial chaos surrogate model of degree two with 38 to 50 experiments, and near-zero probabilities for less than 37 experiments. The rule, deliberately ignored to generate the dataset, is about to be identified empirically by the KBS.



	Nb_Exp	DoE_Type	Meta_Type	PC_Degree	Corr_Func	Scr_Size	Kniter	Trend	Kcrit	Koptalg	Pr
109	[5-15]	halton	Kriging	0	matern7/2	]0-250]	]1000-1500]	linear	ML	BOBYQA	1.0000000
156	[5-15]	sobol	Kriging	0	matern5/2	]250-500]	]0-500]	const	LOO	NelderMead	1.0000000
634	[5-15]	halton	Kriging	0	exponential	]250-500]	]1000-1500]	const	ML	BOBYQA	1.0000000
787	[5-15]	halton	Kriging	0	matern7/2	]250-500]	]500-1000]	linear	ML	NelderMead	1.0000000
794	[5-15]	halton	Kriging	0	matern7/2	]250-500]	]0-500]	linear	LOO	BOBYQA	1.0000000
⋮											
687	[38-50]	sobol	PC	Two	0	0	0	0	0	0	0.201646091
56	[38-50]	halton	PC	Two	0	0	0	0	0	0	0.201244813
251	[38-50]	lhs	PC	Two	0	0	0	0	0	0	0.200000000
712	[38-50]	sobol	PC	Two	0	0	0	0	0	0	0.200000000
718	[16-26]	halton	Kriging	0	matern3/2	]250-500]	]0-500]	const	ML	NelderMead	0.200000000
1778	[38-50]	sobol	PC	Two	0	0	0	0	0	0	0.199596774
1532	[38-50]	halton	PC	Two	0	0	0	0	0	0	0.199143469
179	[38-50]	lhs	PC	Two	0	0	0	0	0	0	0.199124726
⋮											
9	[27-37]	sobol	PC	Two	0	0	0	0	0	0	0
12	[27-37]	lhs	PC	Two	0	0	0	0	0	0	0
26	[27-37]	sobol	Kriging	0	matern3/2	]0-250]	]500-1000]	const	ReML	NelderMead	0
32	[27-37]	lhs	Kriging	0	matern5/2	]0-250]	]0-500]	linear	ReML	BOBYQA	0
40	[27-37]	sobol	Kriging	0	matern7/2	]250-500]	]0-500]	const	ML	BOBYQA	0
49	[27-37]	halton	Kriging	0	matern3/2	0	]1000-1500]	linear	LOO	BOBYQA	0
68	[27-37]	lhs	Kriging	0	gauss	]250-500]	]0-500]	const	ML	NelderMead	0

Figure 10: Excerpt of sorted configurations ranking, predicted by the KBS. Best configurations are on top.

Designers select one of the most efficient configurations, with the lowest number of experiments, for instance the first one. At this stage, users are free to modify the proposed configuration as they wish. The final configuration is defined, as shown in Table 6.

Table 6: Final configuration of the NDoE process.

Variable of the configuration	Value
Objective	Surrogate modelling
Minimal accuracy (Leave-one-out normalized root mean square error, LooNRMSE)	0.05
Minimal predictivity level (Q^2)	0.98
Strategy	Non-adaptive
Numerical model	Rotor1a
Type of NDoE	Halton
Number of experiments	15
Type of surrogate model	Kriging
Kriging model's internal parameters	
Correlation function	Matern 7/2
Trend function	Linear
Optimization criterion	Maximum Likelyhood
Optimization algorithm	BOBYQA
Maximal number of iterations	1500
Initial screening sampling	250

This configuration is set and the NDoE process is executed by Uranie and Code_Aster. As a result, the optimization of the HVAC system relies on a cheaper function. The optimal geometry and material properties of dampers are computed in several minutes instead of hours, but the correctness of the optimal result depends on the quality of the surrogate model.

Then, the performance of this new NDoE process is computed. The kriging surrogate model is as accurate as predicted by the bayesian network. The observed accuracy was 0.011, while the predicted accuracy was between 0.01 and 0.05. But the predictivity of the surrogate model is better than expected. The probability that the value of predictivity was between 0.90 and 0.99 was equal to 0.75 (Table 3). This wrong classification could have been expected, and it will not cause any loss for designers in this case.

This new NDoE process, linked to these new observed results, is capitalized in the knowledge base. Thus, the dataset can continuously grow. The use of this proposed KBS shorten the configuration step of the NDoE process. From thousands possible configurations, the set of relevant configurations was reduced to ten in a very short time. Thus, designers obtained wanted results faster, without using too much computing resources (and then, usable for other design processes managed by other teams) nor loosing time to try and retry different configurations. In this specific case, the surrogate model is obtained earlier. This means simulations on the costly original model of the air-blower system are not required anymore. More simulations can be executed instantaneously, for instance during a project meeting. The product can be optimized earlier in the design process, and the design process itself is shortened. The capitalized knowledge is traced and reusable at will for next NDoE processes, by every designers in the company. Thus, every other projects in the company may profit from this KBS. The human and material resources assigned to the HVAC study can be reallocated earlier to others projects, and the knowledge capitalization can bring help for these other projects.

5 Conclusion

This paper detailed our proposal to make the use of NDoE process more commonly used for product development processes. The solution proposed is based on the capitalization and the management of data and knowledge gathered in organizations. Capitalized knowledge is reused, analyzed to obtain new NDoE process configurations, adapted to a new context. The proposed Knowledge-Based System, using a bayesian network as inference engine, is able to give advice to designers, to integrate existing expert knowledge and to discover new knowledge from these analyses. It is also able to diagnose its abilities for prediction itself. The KBS relies on an ontological model to support the knowledge specific to NDoE process and enhance knowledge transformations processes. The KBS also relies on an inference engine which combine knowledge fast.

Knowledge is continuously updated and capitalized. With the ontological structure, designers can learn from the KBS, and enrich it with their own personal reasoning. The application of this KBS may lead to more efficient design processes of complex products. It also leads to better product quality by controlling the effects of uncertainties, with minimal computational cost, earlier in the design process.

To improve this inference engine, learning process could be enhanced, for instance by applying an hybrid system, or by applying other learning approach, such as unsupervised and reinforcement learning, to be more efficient with a minimal amount of capitalized data. The effect of the number of data (NDoE processes capitalized in the knowledge base) should be studied to evaluate its ability to be use on small knowledge bases. The system is currently set in an automotive company to be validated in an extended enterprise.

Acknowledgments

This work is done in the French FUI project SDM4DOE. We also thank all consortium partners for their contribution during the development of ideas and concepts proposed in the paper.

References

Aamodt, A. and Plaza, E. (1994) 'Case-based reasoning: Foundational issues, methodological variations, and system approaches', *AI communications*, 7(1), pp. 39–59. Available at: http://iospress.metapress.com/index/316258107242JP65.pdf.

Akaike, H. (1970) 'Statistical predictor identification', *Annals of the Institute of Statistical Mathematics*, 22(1), pp. 203–217. doi: 10.1007/BF02506337.

Bayat, S., Cuggia, M., Rossille, D., Kessler, M. and Frimat, L. (2009) 'Comparison of Bayesian network and decision tree methods for predicting access to the renal transplant waiting list.', *Studies in health technology and informatics*, 150, pp. 600–604. doi: 10.3233/978-1-60750-044-5-600.

Beal, A., Claeys-Bruno, M. and Sergent, M. (2014) 'Constructing space-filling designs using an adaptive WSP algorithm for spaces with constraints', *Chemometrics and Intelligent Laboratory Systems*. Elsevier B.V., 133, pp. 84–91. doi: 10.1016/j.chemolab.2013.11.009.

Blondet, G., Le Duigou, J., Boudaoud, N. and Eynard, B. (2015) 'Simulation data management for adaptive design of experiments: A litterature review', *Mechanics & Industry*, 16(6), p. 611. doi:

10.1051/meca/2015041.

Blondet, G., Duigou, J. Le, Boudaoud, N. and Eynard, B. (2018) 'An ontology for numerical design of experiments processes', *Computers in Industry*, 94, pp. 26–40. doi: 10.1016/j.compind.2017.09.005.

Breiman, L. (2001) 'Random Forests', *Machine Learning*, 45(1), pp. 5–32.

Castric, S., Cherfi, Z., Blanchard, G. J. and Boudaoud, N. (2012) 'Modeling Pollutant Emissions of Diesel Engine based on Kriging Models : a Comparison between Geostatistic and Gaussian Process Approach', *14th IFAC Symposium on Information Control Problems in Manufacturing, INCOM'12*. Bucharest, 14(2006), pp. 1708–1715. doi: 10.3182/20120523-3-RO-2023.00038.

Chen, F. T. (1991) 'A personal computer based expert system framework for the design of experiments', *Computers & Industrial Engineering*, 21, pp. 197–200.

Correa, M., Bielza, C. and Pamies-Teixeira, J. (2009) 'Comparison of Bayesian networks and artificial neural networks for quality detection in a machining process', *Expert Systems with Applications*. Elsevier Ltd, 36(3), pp. 7270–7279. doi: 10.1016/j.eswa.2008.09.024.

Cui, C., Hu, M., Weir, J. D. and Wu, T. (2016) 'A recommendation system for meta-modeling: A meta-learning based approach', *Expert Systems with Applications*. Elsevier Ltd, 46, pp. 33–44. doi: 10.1016/j.eswa.2015.10.021.

Dalkir, K. (2005) *Knowledge Management in Theory and Practice*, *Knowledge Management in Theory and Practice*. Elsevier B.V.

Davies, M. (2015) *Knowledge (Explicit, Implicit and Tacit): Philosophical Aspects*. Second Edition, *International Encyclopedia of the Social & Behavioral Sciences: Second Edition*. Second Edition. Elsevier. doi: 10.1016/B978-0-08-097086-8.63043-X.

Denis, J.-B. and Scutari, M. (2014) *Réseaux bayésiens avec R*. EDP Sciences.

Ding, L. and Matthews, J. (2009) 'A contemporary study into the application of neural network techniques employed to automate CAD/CAM integration for die manufacture', *Computers and Industrial Engineering*. Elsevier Ltd, 57(4), pp. 1457–1471. doi: 10.1016/j.cie.2009.01.006.

Dolšak, B. (2002) 'Finite element mesh design expert system', *Knowledge-Based Systems*, 15(5–6), pp. 315–322. doi: 10.1016/S0950-7051(01)00168-X.

Le Duigou, J. and Bernard, A. (2011) 'Product Lifecycle Management Model for Design Information Management in Mechanical Field', in *21st CIRP Design Conference*. Korea, pp. 207–213.

Forrester, A. I. J. and Keane, A. J. (2009) 'Recent advances in surrogate-based optimization', *Progress in Aerospace Sciences*, 45(1–3), pp. 50–79. doi: 10.1016/j.paerosci.2008.11.001.

Garud, S. S., Karimi, I. A. and Kraft, M. (2017) 'Design of computer experiments: A review', *Computers and Chemical Engineering*. Elsevier Ltd, 106, pp. 71–95. doi: 10.1016/j.compchemeng.2017.05.010.

Gaudier, F. (2010) 'URANIE: The CEA/DEN Uncertainty and Sensitivity platform', *Procedia - Social and Behavioral Sciences*. Elsevier Masson SAS, 2(6), pp. 7660–7661. doi: 10.1016/j.sbspro.2010.05.166.

Gorissen, D., Dhaene, T. and Turck, F. De (2009) 'Evolutionary Model Type Selection for Global Surrogate Modeling', *Journal of Machine Learning Research*, 10, pp. 2039–2078.

Hanafy, M. and ElMaraghy, H. (2014) 'Co-design of Products and Systems Using a Bayesian Network', *Procedia CIRP*. Elsevier B.V., 17, pp. 284–289. doi: 10.1016/j.procir.2014.01.129.

Hu, W., Yao, L. G. and Hua, Z. Z. (2008) 'Optimization of sheet metal forming processes by adaptive response surface based on intelligent sampling method', *Journal of Materials Processing Technology*, 197(1), pp. 77–88. doi: 10.1016/j.jmatprotec.2007.06.018.

Iooss, B. and Lemaître, P. (2015) 'A Review on Global Sensitivity Analysis Methods', in Dellino, G. and Meloni, C. (eds) *Uncertainty Management in Simulation-Optimization of Complex Systems-Algorithms and Applications-Part 2*. Springer US, pp. 101–122. doi: 10.1007/978-1-4899-7547-8_5.

El Kadiri, S. and Kiritsis, D. (2015) 'Ontologies in the context of product lifecycle management: state of the art literature review', *International Journal of Production Research*, 53(18), pp. 5657–5668. doi: 10.1080/00207543.2015.1052155.

Kass, G. V. (1980) 'An Exploratory Technique for Investigating Large Quantities of Categorical Data', *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 29, pp. 119–127.

Khan, A. A., Chaudhry, I. A. and Sarosh, A. (2014) 'Case Based Reasoning Support for Adaptive Finite Element Analysis: Mesh Selection for an Integrated System', *Applied Physics Research*, 6(3), pp. 21–39. doi: 10.5539/apr.v6n3p21.

Kiritsis, D. (1995) 'A review of knowledge-based expert systems for process planning. Methods and problems', *The International Journal of Advanced Manufacturing Technology*, 10(4), pp. 240–262. doi: 10.1007/BF01186876.

Lara, J. A., Lizcano, D., Martínez, M. A. and Pazos, J. (2014) 'Data preparation for KDD through automatic reasoning based on description logic', *Information Systems*. Elsevier, 44, pp. 54–72. doi: 10.1016/j.is.2014.03.002.

Liu, H., Zhou, S., Lam, W. and Guan, J. (2017) 'A new hybrid method for learning bayesian networks: Separation and reunion', *Knowledge-Based Systems*. Elsevier B.V., 121, pp. 185–197. doi: 10.1016/j.knosys.2017.01.029.

Lopez de Mantaras, R., McSherry, D., Bridge, D., Leake, D., Smyth, B., Craw, S., Faltings, B., Maher, M. Lou, Cox, M. T., Forbus, K., Keane, M., Aamodt, A. and Watson, I. (2005) 'Retrieval, reuse, revision and retention in case-based reasoning', *The Knowledge Engineering Review*, 20(03), p. 215. doi: 10.1017/S0269888906000646.

Lorenzen, T. J., Truss, L. T., Spangler, W. S., Corpus, W. T. and Parker, A. B. (1992) 'DEXPERT: an expert system for the design of experiments', *Statistics and Computing*, 2(2), pp. 47–54. doi: 10.1007/BF01889582.

Madsen, A. L., Jensen, F., Salmeron, A., Langseth, H. and Nielsen, T. D. (2015) 'Parallelisation of the PC Algorithm', in Puerta, J. M., Gámez, J. A., Dorronsoro, B., Barrenechea, E., Troncoso, A., Baruque, B., and Galar, M. (eds) *Advances in Artificial Intelligence*. Cham: Springer International Publishing (Lecture Notes in Computer Science), pp. 14–24. doi: 10.1007/978-3-319-24598-0.

Naïm, P., Wuillemin, P.-H., Leray, P., Pourret, O. and Becker, A. (2007) *Réseaux bayésiens*. 3e édition. Eyrolles.

Naranje, V. and Kumar, S. (2014) 'A knowledge based system for automated design of deep drawing die for axisymmetric parts', *Expert Systems with Applications*. Elsevier Ltd, 41(4), pp. 1419–1431. doi: 10.1016/j.eswa.2013.08.041.

Nonaka, I., Toyama, R. and Byosière, P. (2001) 'A Theory of Organizational Knowledge Creation: Understanding the Dynamic Process of Creating Knowledge', in *Handbook of Organizational Learning and Knowledge*. New York, NY: Elsevier, pp. 491–517. Available at:

http://linkinghub.elsevier.com/retrieve/pii/B9780750670098500161.

Oishi, A. and Yagawa, G. (2017) 'Computational mechanics enhanced by deep learning', *Computer Methods in Applied Mechanics and Engineering*. Elsevier B.V., 327, pp. 327–351. doi: 10.1016/j.cma.2017.08.040.

Patelli, E., Murat Panayirci, H., Broggi, M., Goller, B., Beaurepaire, P., Pradlwarter, H. J. and Schuëller, G. I. (2012) 'General purpose software for efficient uncertainty management of large finite element models', *Finite Elements in Analysis and Design*, 51, pp. 31–48. doi: 10.1016/j.finel.2011.11.003.

Pearl, J. (2000) *Causality: Models, Reasoning and Inference*. Edited by Cambridge University Press. Cambridge, England.

Poeschl, S., Lieb, J., Wirth, F. and Bauernhansl, T. (2017) 'Expert Systems in Special Machinery: Increasing the Productivity of Processes in Commissioning', in *The 50th CIRP Conference on Manufacturing Systems*, pp. 545–550. doi: 10.1016/j.procir.2017.03.162.

Powers, D. M. W. (2011) 'Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.', *Journal of Machine Learning Technologies*, 2(1), pp. 37–63.

Quinlan, J. R. (1996) 'Improved use of continuous attributes in C4.5', *Journal of artificial intelligence research*, 4, pp. 77–90.

Russell, S. (2010) *Artificial Intelligence A Modern Approach*. 3rd edn. Edited by Pearson.

Sahin, S., Tolun, M. R. and Hassanpour, R. (2012) 'Hybrid expert systems: A survey of current approaches and applications', *Expert Systems with Applications*. Elsevier Ltd, 39(4), pp. 4609–4617. doi: 10.1016/j.eswa.2011.08.130.

Sakthivel, N. R., Sugumaran, V. and Babudevasenapati, S. (2010) 'Vibration based fault diagnosis of monoblock centrifugal pump using decision tree', *Expert Systems with Applications*. Elsevier Ltd, 37(6), pp. 4040–4049. doi: 10.1016/j.eswa.2009.10.002.

Salehi, H. and Burgueño, R. (2018) 'Emerging artificial intelligence methods in structural engineering', *Engineering Structures*. Elsevier, 171(May), pp. 170–189. doi: 10.1016/j.engstruct.2018.05.084.

Sanchez, S. M. and Wan, H. (2012) 'Work smarter, not harder: a tutorial on designing and conducting simulation experiments.', in Laroque, C., Himmelspach, J., Pasupath, R., Rose, O., and Uhrmacher, A. M. (eds) *Winter Simulation Conference*. Berlin, Germany: IEEE, pp. 1929–1943.

Sarrafzadeh, M., Martin, B. and Hazeri, A. (2006) 'LIS professionals and knowledge management: some recent perspectives', *Library Management*, 27(9), pp. 621–635. doi: 10.1108/01435120610715527.

Schmidhuber, J. (2015) 'Deep Learning in neural networks: An overview', *Neural Networks*. Elsevier Ltd, 61, pp. 85–117. doi: 10.1016/j.neunet.2014.09.003.

Schwarz, G. (1978) 'Estimating the dimension of a model', *The Annals of Statistics*, 6(2), pp. 461–464. Available at: http://projecteuclid.org/euclid.aos/1176345976.

Scutari, M. (2010) 'Learning Bayesian Networks with the bnlearn R Package', *Journal of Statistical Software*, 35(3), pp. 1–22. doi: 10.18637/jss.v035.i03.

Scutari, M. (2014) 'Bayesian Network Constraint-Based Structure Learning Algorithms: Parallel and Optimised Implementations in the bnlearn R Package', *arXiv.org*. Available at: http://arxiv.org/abs/1406.7648%5Cnpapers2://publication/uuid/45ECC151-D998-4AC2-9449-

41770936B214.

Shraim, M. S. (1989) *An expert system for designing statistical experiments*. Ohio University.

Slanzi, D. and Poli, I. (2014) 'Evolutionary Bayesian Network design for high dimensional experiments', *Chemometrics and Intelligent Laboratory Systems*. Elsevier B.V., 135, pp. 172–182. doi: 10.1016/j.chemolab.2014.04.013.

Soldatova, L. N. and King, R. D. (2006) 'An ontology of scientific experiments', *Journal of The Royal Society Interface*, 3(11), pp. 795–803. doi: 10.1098/rsif.2006.0134.

Spirtes, P., Glymour, C. and Scheines, R. (1993) *Causation, Prediction, and Search*. Edited by Springer-Verlag. New York, NY: Springer New York (Lecture Notes in Statistics). doi: 10.1007/978-1-4612-2748-9.

Sugumaran, V. and Ramachandran, K. I. I. (2011) 'Effect of number of features on classification of roller bearing faults using SVM and PSVM', *Expert Systems with Applications*. Elsevier Ltd, 38(4), pp. 4088–4096. doi: 10.1016/j.eswa.2010.09.072.

Tasdemir, S., Saritas, I., Ciniviz, M. and Allahverdi, N. (2011) 'Artificial neural network and fuzzy expert system comparison for prediction of performance and emission parameters on a gasoline engine', *Expert Systems with Applications*. Elsevier Ltd, 38(11), pp. 13912–13923. doi: 10.1016/j.eswa.2011.04.198.

Tkáč, M. and Verner, R. (2016) *Artificial neural networks in business: Two decades of research*, *Applied Soft Computing Journal*. Elsevier B.V. doi: 10.1016/j.asoc.2015.09.040.

Urrea, C., Henríquez, G. and Jamett, M. (2015) 'Development of an expert system to select materials for the main structure of a transfer crane designed for disabled people', *Expert Systems with Applications*. Elsevier Ltd, 42(1), pp. 691–697. doi: 10.1016/j.eswa.2014.08.017.

Vanschoren, J. and Soldatova, L. N. (2010) 'Exposé: An ontology for data mining experiments', in *Proc. of the 3rd Int. Workshop on Third Generation Data Mining: Towards Service-oriented Knowledge Discovery (SoKD)*. Barcelona, Spain, pp. 31–46.

Wagner, W. P. (2017) 'Trends in expert system development: A longitudinal content analysis of over thirty years of expert system case studies', *Expert Systems with Applications*. Elsevier Ltd, 76, pp. 85–96. doi: 10.1016/j.eswa.2017.01.028.

Weiner, D. J. (1992) 'Expert systems to aid in the formulation of hypotheses and design of experiments in biomedical research', *Mathematical and Computer Modelling*, 16(6–7), pp. 185–198. doi: 10.1016/0895-7177(92)90162-E.

Yondo, R., Andrés, E. and Valero, E. (2018) 'A review on design of experiments and surrogate models in aircraft real-time and many-query aerodynamic analyses', *Progress in Aerospace Sciences*, 96(December 2017), pp. 23–61. doi: 10.1016/j.paerosci.2017.11.003.