Industrial Management & Data Systems

Design freeze sequencing using Bayesian network framework

Jihwan Lee Yoo S. Hong

Design freeze sequencing using Bayesian network framework

Jihwan Lee and Yoo S. Hong
Department of Industrial Engineering, Seoul National University, Seoul, Republic of Korea

Received 25 March 2015
Revised 11 June 2015
Accepted 13 June 2015

Abstract

Purpose – Change propagation is the major source of schedule delays and cost overruns in design projects. One way to mitigate the risk of change propagation is to impose a design freeze on components at some point prior to completion of the process. The purpose of this paper is to propose a model-driven approach to optimal freeze sequence identification based on change propagation risk.
Design/methodology/approach – A dynamic Bayesian network was used to represent the change propagation process within a system. According to the model, when a freeze decision is made with respect to a component, a probabilistic inference algorithm within the Bayesian network updates the uncertain state of each component. Based on this mechanism, a set of algorithm was developed to derive optimal freeze sequence.
Findings – The authors derived the optimal freeze sequence of a helicopter design project from real product development process. The experimental result showed that our proposed method can significantly improve the effectiveness of freeze sequencing compared with arbitrary freeze sequencing.
Originality/value – The methodology identifies the optimal sequence for resolution of entire-system uncertainty in the most effective manner. This mechanism, in progressively updating the state of each component, enables an analyzer to continuously evaluate the effectiveness of the freeze sequence.
Keywords Design process, Change propagation, Bayesian network, Design freeze, Freeze sequence
Paper type Research paper

1. Introduction

Many product design processes are large and interdisciplinary in nature. A product usually consists of a set of components each of which is designed by a separate group of engineers (Eppinger *et al.*, 1994). Complex systems such as automobiles or aircraft can involve even thousands of engineers making millions of design decisions over the course of years. As a product becomes complex and comes to involve many more decision makers, coordinating the design of components grows very complicated. Management of complexity within the design and development process, not surprisingly, is a frequent focus of engineering management research.

A major aspect of design complexity, namely change propagation, is considered a major threat to the efficiency and effectiveness of the product development process (Clarkson *et al.*, 2004). Engineers must continuously modify their design in response to changes that are either exogenous (e.g. shifting of customer needs) or endogenous (e.g. the discovery of a better solution approach) to the design process. Change propagation occurs due to design dependencies between components. Since the design parameters of a component interact with those of other components, a change made to

This research was supported by the Basic Science Research Program, administered by the National Research Foundation of Korea (NRF) and funded by the Ministry of Education, Science and Technology (NRF-2012R1A1A2005995). The authors deeply appreciate the administrative support during the project period from Engineering Research Institute (ERI) of Seoul National University.

Industrial Management & Data Systems
Vol. 115 No. 7, 2015
pp. 1204-1224

0263-5577
DOI 10.1108/IMDS-03-2015-0095

one component can necessitate one or more additional changes to the system (Giffin *et al.*, 2009). Sometimes, with especially complex designs, a change might propagate throughout the entire system, incurring significant development cost and schedule delay.

One way of mitigating the risk of change propagation is successive freezing of components (Thomke, 1997). Design freeze is defined as the end point of the design phase at which a technical product description is handed over to production. By freezing some components earlier, company can expect two main advantages: first, overall process can be quickly stabilized by limiting the engineering changes on that component; and second, since an early frozen component can be handed over to the manufacturing phase in advance, it can reduce overall product development lead-time. However, at the same time, an early-frozen component might still be vulnerable to changes propagated from other components. It is not uncommon that, due to component interdependency, an early frozen component has to be redesigned, which can lead to significant rework costs. In this light, careful planning of design freeze sequence is required for efficient management of change propagation during design process.

This paper proposes a model for change-propagation-risk-based determination of optimal freeze order. In order to identify the optimal freeze sequence, the Bayesian network (BN), which is an emerging tool for a wide range of risk management tasks, is used as a modeling framework for representing a sequential freeze process. When a freeze decision is made with respect to one component, the change propagation risk associated with it is removed from the system. In this setting, the optimal freeze sequence is that which reduces risk to the system in most effective manner.

2. Related study

2.1 Design freeze

One commonly perceived viewpoint about design process is an uncertainty reduction process, where the design description begins as a vague concept and gradually reduces the solution space until a precise final solution is reached (Herrmann, 2010). From this view point, a design for a system or a component cannot be made in a single step. Rather, it can be considered as a progressive process in which parameters are incrementally defined and frozen (Maier *et al.*, 2014). Design freeze is defined as binding decision that defines the whole product, its parts or parameters and allows the continuation of the design based on that decision (Eger *et al.*, 2005). In engineering management literature, design freeze is considered as a strategy for accelerating the product development time (Zirger and Hartley, 1996). A number of mathematical models has been proposed to determine timing for design freeze. Krishnan *et al.* (1997) evaluates a trade-off between early and late freeze timing between upstream and downstream design tasks. Bhattacharya *et al.* (1998) proposes a mathematical model for determining design freeze timing in the presence of competitors and market uncertainty. Huchzermeier and Loch (2001) proposes a real-option model which can evaluate the flexibility of design freeze timing under various types of product risk.

Although these models shed a light on the best timing of design freeze, most of them relied on a simple assumption that there is a single design freeze point during the design process. Based on the case studies of many engineering companies, however, Eger *et al.* (2005) found that individual parts are actually frozen at different times. At the part level, engineers sequentially freeze components in order to reduce the likelihood of further engineering changes. By freezing some components earlier than others, the designers can reduce the likelihood of change propagation and facilitate a design continuation of dependent components. This, in turn, calls for a structured method for deriving careful freeze sequence of components.

Motivated by Eger *et al.*'s (2005) work, Keller *et al.* (2008) proposes a structured method for determining design freeze sequence of components. They used Clarkson *et al.* (2004)'s design structure matrix (DSM)-based method to calculate the combined risk of change propagation of each component. Then, a heuristic optimization method, simulation annealing, is used to reshuffle the rows and columns of matrix in order to freeze more influential components as early as possible. Although change prediction methods (CPM) is a good tool for assessing the risk of change propagation, it cannot directly take into account the freeze states of each part, which makes it impossible to address the dynamic evolution of component states during the freeze process. The present study adopts BN to overcome this limitation by modeling probabilistic relationship among components.

2.2 Change propagation analysis techniques

Prediction of engineering changes is a major research area in engineering change management. A number of tools have been developed in this domain to anticipate the impact of change propagation. One of the first models is Change Favorable Representation (C-FAR) proposed by Cohen *et al.* (2000). C-FAR quantitatively measures dependency between design parameters and proposes a mechanism for calculating the cascading effect of change propagation. Clarkson *et al.* (2004) proposed a more advanced tool named CPM. The distinctive feature of CPM is that it predicts the impact of change propagation with the risk measure which is obtained by multiplying the likelihood and impact. CPM quantifies the likelihood and impact between adjacent components using a DSM. Then, an algorithm, which enumerates every possible change propagation path derived with the DSM, calculates each component's expected risk. After its successful initial implementation, CPM has been further extended by numerous studies. For example, Keller *et al.* (2005) proposed a data visualization tool for CPM. More recently, Koh *et al.* (2012), extending DSM tomulti-domain matrix, addressed change propagation among different domains such as organizations or manufacturing processes.

In contrast, relatively little research has been done for the management of engineering changes during the product design process. Oh *et al.* (2007) uses change propagation information to design a system architecture which can effectively absorb change propagation. Wynn *et al.* (2010) and Maier *et al.* (2014) proposes methodologies for prioritizing design activities considering lead-time delays caused by change propagation between components. Yang and Duan (2012) develops a parameter linkage model which represents the propagation of change at the design parameter level, and proposes a methodology for searching an optimal change propagation path which can maximally mitigate the effect of propagation. Our design freeze sequencing method falls into this stream of research in that it provides a dynamic way to manage the propagation of change during the design phase. For more comprehensive literature review about change propagation analysis, please refer to Hamraz *et al.* (2013) and Jarratt *et al.* (2011).

2.3 BN

A BN is a directed acyclic graph (DAG) in which the nodes represent the system variables and the arcs symbolize the dependencies or the cause-effect relationships among the variables. Due to its ability to compactly represent dependence and independence relationship among random variables, it has been used as a robust and efficient framework for modeling and reasoning uncertain knowledge which is often represented by large number of variables (Kjaerulff and Madsen, 2008) including wide range of applications, such as fault detection (Bobbio *et al.*, 2001), operational risk management (Cowell *et al.*, 2007), or medical diagnosis (Heckerman and Nathwani, 1992). For more details about the real world applications of BN, refer to Heckerman *et al.* (1995b).

Over the last decades, a number of papers in engineering design domain has adopted BN in solving design problems. Since BN is a useful tool for modeling uncertainty, most of them use the formalism of BN in design decision making under uncertainty. For example, Matthieu *et al.* (2012) uses BN in formulating optimal disassembly strategy considering both product architecture and quality uncertainty. Moullec *et al.* (2013) and Shahan and Seepersad (2012) use the formalism of BN in representing probabilistic relationship between design parameters and predict the probability distribution of product performance. Matthews (2011) develops a BN-based concept design support system which provides dynamic guides for selecting design elements within the morphological chart. However, the application of BN in engineering change management has not been reported in literature. Although Morkos *et al.* (2014) proposes a neural network in predicting engineering changes, their tool does not explicitly address the propagative property of engineering changes.

In the present study, we address a special type of decision problem, namely the sequential decision problem (Mookerjee and Mannino, 1997). In this problem, the decision maker has to choose among a set of alternative actions. To maximize his utility, the decision maker can sequentially gather new information through a series of tests. The objective of this problem, thus, is to choose the right test to perform next. BN is especially useful in evaluating the value of information of a sequence of testing because it can compute the probability distributions for a set of variables based on the observation of those variables (Mussi, 2002). A number of papers on the application of sequential decision problem using BN have been reported. Heckerman *et al.* (1995a) proposes an algorithm for deriving an optimal troubleshooting sequence from a BN model which models the relationship between the failure modes and their associated components. Similarly, Huang *et al.* (2008) proposes a method for deriving the sequence of diagnosis for automobile sound systems, Skaanning *et al.* (2000) a system for trouble shooting printers, Mirarab and Tahvildari (2007) an optimal test sequence for software systems, etc. Vomlel (2004) applies a similar approach in educational testing.

In the present study, the framework of sequential decision problem is adopted to obtain design freeze sequence. Probabilistic relationships between components are represented with BN. When a component is frozen, the information about this component is updated throughout the network since the associated uncertainty is resolved. Each time a component is frozen, thus, we can dynamically update the uncertainty levels of each component. In this fashion, we can quantitatively evaluate the freeze sequences, which was not properly handled in previous literature.

3. Problem definition

The problem of finding the optimal freeze sequence can be formulated as follows:

- given: C, a set of components, σ_C, the set of permutations of C, and f, *a* function that evaluates σ_C; and
- problem: find $S' \in \sigma_C$ such that $f(S') \geqslant f(S'')$ *for* $\forall S''$ $(S'' \in \sigma_C, S'' \neq S)$.

Here, σ_C represents the set of all possible freeze sequences, and f is the function that evaluates the effectiveness of freeze sequence. Thus, the problem is to choose the best sequence of components maximizing the user-defined evaluation function f. To further define our problem as a sequential decision problem, the following assumptions are required:

- Sequential freeze process: decision maker freezes one component at a time.
- Initial change probability is identified for each component: represents the probability that a change first arrives at that component. Change can occur due to safety issues, new technical solutions, or a change of customer request. Note, however, that this probability only describes the initiation of changes. The component might have a probability higher than the initial one, due to change propagation from other components.
- Change propagation probability is identified among components: represents the probability that a change that appears in one component results in changes to the other components.

When a component is frozen, it does not initiate change propagation to other components, therefore, decrease the change propagation risk of entire system. Figure 1 illustrates a trajectory of change propagation risk of a design freeze sequence of five components. As can be seen, each point represent the design freeze point and the values assigned over each point is average residual risk of unfrozen components. BN is utilized in calculating dynamic evolution of change propagation risk given a current freeze sequence. As more components are frozen, the design begins to stabilize, and when the final component is frozen, the change probability of each component falls to zero. In this setting, our objective is to find optimal design freeze sequence which can maximally mitigate the change propagation during design process. For, the rest of the paper, we would discuss more detail about this process.

4. Modeling freeze process using BN

4.1 Introduction to BN

BN construction requires both qualitative and quantitative parts. The qualitative part is represented by a graph. The nodes in the graph represent a set of random variables,

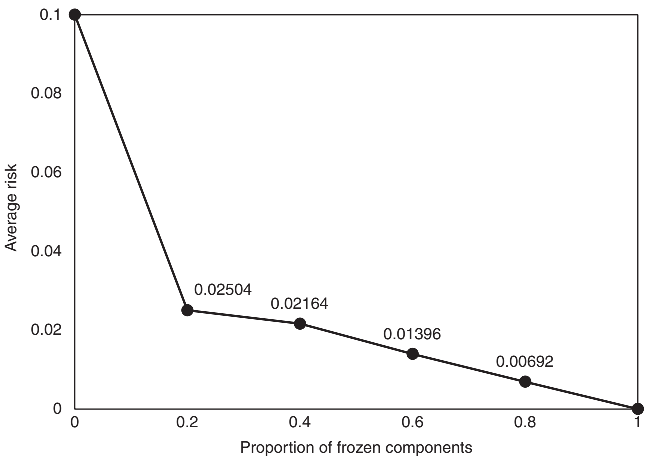


Figure 1. Example of change propagation risk trajectory

$X = \{X_1, X_2, ..., X_n\}$, from a domain. A set of directed edge (or arc) connecting pairs of nodes $X_i \rightarrow X_j$ represents the direct dependency between random variables, indicating that X_i is the direct cause of X_j. The only constraint on the arc is that it should not make any directed cycle in the graph. Therefore, whenever the DAG assumption is maintained, any kind of cause-effect relationship among random variables can be encoded in the BN.

For the quantitative part, the strength of the probabilistic relationship between random variables is assigned by a conditional probability table (CPT). Assuming a discrete random variable within the network, the CPT contains a set of conditional probability distributions for every possible instantiation of its parents. An example of a BN is provided in Figure 2. For example, assuming binary state of each component, the CPT of C consists of four distinctive conditional probability distributions for the combination A and B.

Once the graph and CPTs have been identified, they can be used to calculate the complete joint probability distribution. The BN utilizes the structural property encoded in G to reduce the computational burden of calculating the full joint probability distribution, which otherwise grows exponentially with the number of random variables. This property, named d-separation, enables the full joint probability distribution of $X = \{X_1, ..., X_n\}$ to be factorized as in Equation (1), which often is referred to chain-rule:

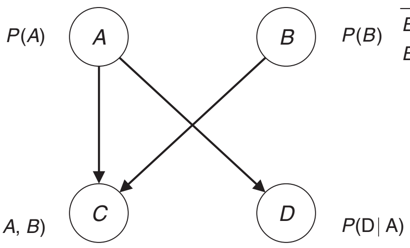
$$P(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} P\left(X_i \middle| X_{pa(i)}\right). \tag{1}$$

4.2 BN-based change propagation modeling

4.2.1 Modeling of cause-effect relationship among components. In order to sequentially calculate the uncertainty of each component given freeze states of components, a BN is proposed. In this BN, the node corresponds to the component of which design can be frozen by a separate group of engineers. This node is a discrete and binary random variable the state of which takes "yes" if it accepts a design change, and "no" otherwise. Edges connecting two nodes indicate direct dependency between components.

To represent cascading effect of changes which occurs through several intermediates steps, each node should be extended with temporal dimension. This

A = true	0.3
A = false	0.7



B = true	0.5
B = false	0.5

	A = true		A = false	
	B = true	B = false	B = true	B = false
C = true	0.9	0.1	0.95	0.12
C = false	0.1	0.9	0.05	0.88

	A = true	A = false
D = true	0.8	0.3
D = false	0.2	0.7

Figure 2.
Example BN with four random variables

types of BN is often called dynamic Bayesian network (DBN). The typical structure of the DBN is illustrated in Figure 3. It consists of a sequence of sub-models, each of which represents the state of the system at a certain point in time, which representation is called a time point. Temporal edges connecting nodes between consecutive sub-models reveal such temporal dependency between states. Although a DBN models a dynamic system, its structure is time-invariant; that is, the structure of the network does not change over time, with the exception of the root node (i.e. the node at time period 1). Therefore, whenever the prior distribution of the root node is specified, the DBN can recursively update the network, enabling a user to predict the further behavior of the system for the desired number of iterations.

As an illustrative example, the DBN of change propagation process among five components is illustrated in Figure 4. The temporal node c_i^t represents the uncertain state of component i at time period t. Each time period indicates an intermediate change propagation step during the change propagation process. If a change arrives in the root nodes, it can cascade through the change propagation path revealed by temporal edges $c_i^{t-1} \rightarrow c_j^t$ connecting nodes between consecutive stages. For example, in Figure 4, if a change arrives in c_c^1, this change can propagate c_d^2. Again, the change of c_d^2 can

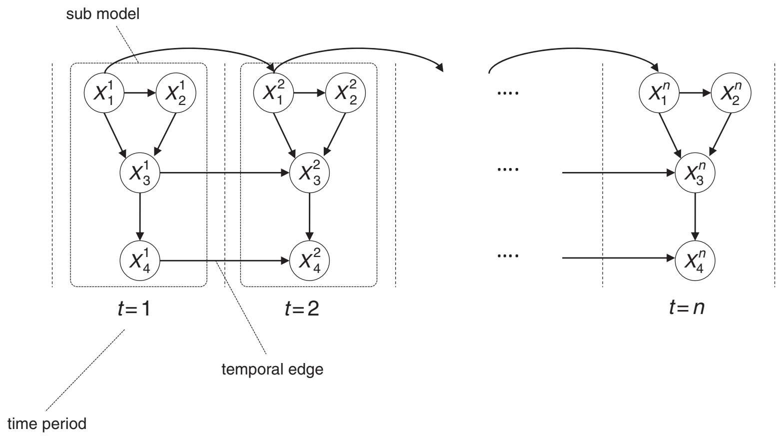


Figure 3.
Typical structure of DBN

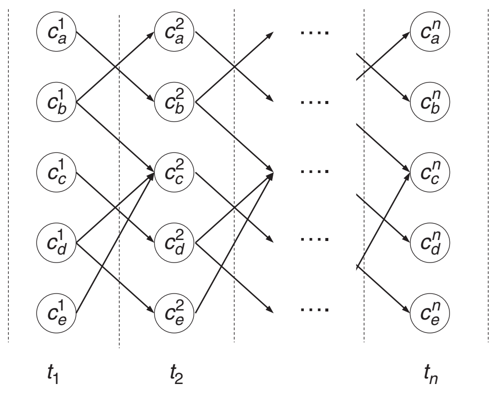


Figure 4.
Change propagation network of five components

result in changes to c_e^3 and c_c^3, respectively. In this way, a user can unroll the change propagation process for the desired number of time stages, duplicating a set of nodes and temporal edges. However, it is noteworthy that temporal edges connecting nodes between consecutive time-slices do not change during the change propagation process, since the structure of the DBN is time-invariant.

4.2.2 Assessment of change propagation probability. In order to quantify the degree of change propagation between components, each node should be identified with the CPT. For the root node, only the prior probability is required. This probability represents the arrivals of engineering changes to the component. Meanwhile, the nodes in the intermediate time periods (from 2 to n) describes the behaviors of the change propagation process. Since these nodes are influenced by their immediate predecessor, the conditional probability distribution given every instantiation of their parents should be specified. Consider, again, the CPT of c_b^2 in Figure 4. As can be seen, the state of c_b^2 is affected by c_a^1. Since c_b^2 does not change without the change of c_a^1, the conditional probability $p\left(c_b^2 = \text{yes}\middle|c_a^1 = \text{no}\right)$ becomes zero. In contrast, c_b^2 might change with some probability if the state of c_a^1 is set to yes. Suppose that such chance probability is 30 percent. Then, the CPT of c_b^2 can be defined as follows:

$$\begin{aligned} &p(c_j^t = \text{yes}|c_i^{t-1} = \text{no}) = 0\\ &p(c_j^t = \text{no}|c_i^{t-1} = \text{no}) = 1\\ &p(c_j^t = \text{yes}|c_i^{t-1} = \text{yes}) = 0.3\\ &p(c_j^t = \text{no}|c_i^{t-1} = \text{yes}) = 1{-}0.3 \end{aligned} \tag{2}$$

Since the DBN has a time-invariant structure, each c_b^t during intermediates states has the same CPT as that of c_b^2.

When a component is connected by many components, the number of parameters to be estimated increases exponentially; so, if n parents is connected to a common child, then the number of probability distributions to be estimated is 2^n. Therefore, as n increases, identifying the conditional probability distribution given every parent combination becomes computationally burdensome.

The knowledge elicitation burden of constructing the CPT can be reduced by assuming a conditional independence relationship among parents. Suppose that there are several causes $X_1, X_2, \ldots, X_n$ and a common effect variable Y, where each of the causes X_i has the probability p_i of being sufficient to produce the effect. The conditional independence among $X_1, X_2, \ldots, X_n$ with respect to Y holds when each parent's ability to produce the effect Y is not influenced by the presence of other parents. When conditional independence is assumed, the CPT of Y can be obtained by the equation:

$$p\left(y\middle|X_p\right) = 1 - \prod_{i:X_i \in X_p} (1-p_i) \tag{3}$$

where X_p is the set of every instantiation of its parent. This special structure of CPT is referred to as the Noisy-OR model. For example, consider the example of c_b^2 in Figure 4, which is affected by both c_d^1 and c_e^1. The CPT of c_b^2 under Noisy-OR model is depicted in Table I. As can be seen, if we know the change propagation probability of

$p\left(c_b^2 = \text{yes} \middle| c_d^1 = \text{yes}\right)$ and $p\left(c_b^2 = \text{yes} \middle| c_e^1 = \text{yes}\right)$, respectively, then, $p\left(c_b^2 = \text{yes} \middle| c_d^1 = \text{yes}, c_e^1 = \text{yes}\right)$ can be calculated without further knowledge elicitation using Equation (3).

4.3 Calculation of change propagation risk given design freeze decision

After BN is constructed, it can be used to answer various probabilistic queries. When a certain state of a node is observed by a decision maker, this node is called "evidence." This evidence e, then, propagates across the network, updating a new posterior probability distribution $p(X|e)$ for each variable. The BN provides a mechanism for calculating the posterior probability distribution of a certain hypothetical variable x given the availability of a set of evidence e. This $P(x|e)$-calculation task is often called probabilistic inference.

The concept of probabilistic inference is used to update the change propagation risk as design freeze sequence progress. Suppose that we want to calculate the remaining risk of each component after component a is frozen in Figure 4. Since the frozen component does not allow changes, the true state of the corresponding root node c_a^1 is set to "no" regardless of the external event. Now the posterior probability distribution $p\left(c_x = \text{yes} \middle| c_a^1 = \text{yes}\right)$ indicates the remaining risk to all five components. We can then proceed to the next component by making additional observations on the root nodes. The change propagation risk of freezing component b after freezing component a, then, can be calculated by computing $p\left(c_x \middle| c_a^1 = \text{no}, c_b^1 = \text{no}\right)$. In this way, we can update the change propagation risk of every component until the final component is frozen. By means of this scheme, we can compare the effectiveness of all freeze sequence alternatives and identify the optimal one among them.

5. Derivation of optimal freeze sequence

Finding optimal freeze sequence can be viewed as finding the optimal sequence of making evidence on root node. Such a sequential decision problem is difficult to solve, however, because there is no simple closed-form solution that has been found. Therefore, one must either check all possible sequences or use a heuristic to check, based on a greedy approach. In the following section, we will introduce several algorithms for identifying the component freeze order.

5.1 All enumeration algorithm

An algorithm that can always guarantee the optimal freeze sequence needs to consider all possible freeze sequence orderings. The all enumeration algorithm, correspondingly, identifies the optimal freeze order by searching all possible sequences. When a product consists of n components, each of the n! orderings is checked in order to determine the optimal order. It is clear that the freeze-sequencing problem is a subset of the traveling

Table I.
Conditional probability table of component c

Component d	Component e	Yes	No
Yes	Yes	1−(1−0.3)(1−0.5) = 0.85	(1−0.3)(1−0.5) = 0.15
Yes	No	1−(1−0.3) = 0.5	(1−0.3) = 0.5
No	Yes	1−(1−0.3) = 0.3	(1−0.3) = 0.7
No	No	0	1

Source: Based on Figure 5

salesman problem, a famous NP-hard problem, even if its computation procedure is deterministic. The all enumeration algorithm can be described as follows.

All enumeration algorithm:

Step 1: Extract initial sequence from the permutation set

$S' \leftarrow$ the initial sequence from the permutation set σ_C

$$\sigma_C \leftarrow \sigma_C \backslash \{S'\}$$

Step 2: Proceed to next sequence and extract it from the permutation set

$S'' \leftarrow$ next sequence to consider

$$\sigma_C \leftarrow \sigma_C \backslash \{S''\}$$

If $\sigma_C = \{\varnothing\}$ then terminate.

Step 3: Compare sequence

If $f(S') \leqslant f(S'')$ then $S' \leftarrow S''$

Otherwise remains same.

Step 4: Go back to step 2.

As an illustrative example, the all enumeration algorithm was applied to the five-component example from Figure 1. However, since the inherent complexity of probabilistic inference is NP-hard, when the size of the network increases, searching the optimal sequence from among all possible combinations becomes intractable.

5.2 Myopic search algorithm

When solving a sequential decision problem, a myopic (greedy) algorithm can be utilized as a good approximation (Li *et al.*, 2007). The underlying strategy of the myopic search algorithm is to always choose the best option given the current situation. In our problem, the myopic approach was applied such that the component that can maximally mitigate the overall risk is frozen first, followed by the second best component, and so on until to the final component. Although the solution optioned by the myopic search might be suboptimal, it can reduce the complexity of searching sequence alternatives to n without any significant loss of accuracy. The myopic algorithm can be described as follows.

Myopic search algorithm:

Step 1: Initialize remaining component K and ordered set of optimal sequences J

Step 2: Identify components that can maximally mitigate change propagation risk R

Compute $\arg\min_x R$

$$J \leftarrow J \cup \{x\}$$

$$K \leftarrow K - \{x\}$$

If $J = \varnothing$ then Stop.

Step 3: Go to step 2.

If every component is assigned a uniform cost, the myopic search algorithm can guarantee an optimal solution, because the trajectory of change propagation risk always shows non-increasing patterns. However, if the cost of every component is different, the myopic search algorithm cannot guarantee an optimal solution.

5.3 Non-myopic search algorithm

The myopic search algorithm looks only at the effect of freezing one component at a time. As such, this approach cannot guarantee the optimality of the solution. In order to improve the solution without sacrificing too much complexity, a non-myopic algorithm can be utilized.

One promising approach is to use the K-optimal algorithm. The principle of this algorithm is simple: it enumerates K-pairs of decision elements together, and selects the best pairs for each algorithm cycle. In our problem, the K-optimal algorithm, rather than search one component at a time, freezes K-pairs of components. After freezing the best component pairs, the same procedure is applied to the remaining components. The K-optimal algorithm can be described as follows.

K-optimal algorithm:

Step 1: Initialize remaining component K and ordered set of optimal sequences J.

Step 2: Enumerate all possible k-pairs of components from K. Denote this set as S_k.

Step 3: Identify best component pairs that can maximally mitigate change propagation risk R.

Compute $s = \arg\min_{S_k} \mathrm{R}$

Step 4: Re-order s such that it can maximally mitigate change propagation risk r.

Step 5: Update freeze sequence J and remaining component K:

$$J \leftarrow J \cup \{s\}$$

$$K \leftarrow K - \{s\}$$

If $K = \varnothing$ then Stop.

Step 6: Go to step 2.

Although solution quality improves with increasing value of K, computation time also increases, due to the increased number of component pairs. Johnson and McGeoch (1997) showed that for $k > 3$, the computation time increases considerably faster than the solution quality, indicating that the two-optimal approach is both fast and effective.

6. Case study

6.1 Product descriptions

For illustration, our model is applied to the product data of Westland Helicopter EH101, which was originally obtained by Clarkson *et al.* (2004). They obtain this data through workshops and interviews of component designers. They used DSM to decompose the helicopter into subsystems, such as engines, weapons, or avionics, etc. Figure 5 shows the DSM of EH101. The (i, j) element in the matrix indicates the change propagation probability from component i to component j. As illustrated, this helicopter consists of 19 subsystems, the components of which are interrelated with complex interdependency.

6.2 DBN model representation

First, the DSM information is converted into our BN-based change propagation model. This process is straightforward. Each component in the DSM is converted to a node in the graph. Edges can be easily identified by referring the dependency structure in the DSM. After the graph structure is identified, the CPT of each node should be identified. In case the number of parents is too many, eliciting change probabilities given every

Air_conditioning					0.2	0.2		0.2		0.2		0.2	0.2						
Auxiliary_electrics	0.2			0.2	0.2	0.5				0.2						0.2		0.2	
Hydraulics						0.2			0.2								0.2	0.2	0.2
Ice_and_rain_protection												0.5			0.2		0.2	0.2	
Avionics	0.2		0.2	0.2		0.5	0.2	0.5	0.5			0.7		0.2	0.5			0.2	0.5
Fuselage_additional_items										0.2									
Fuel						0.2		0.2		0.2		0.7							
Engine_auxiliaries									0.2			0.7							
Flight_control_system			0.2		0.2			0.2				0.2			0.2	0.2	0.2	0.2	0.2
Bare_fuselage	0.7	0.2	0.2	0.2	0.7	0.7	0.5		0.2		0.7	0.7	0.7	0.7	0.2		0.2	0.2	0.5
Cabling_and_piping	0.5	0.7	0.5	0.5	0.7	0.5	0.5	0.2	0.5	0.5		0.7		0.7				0.5	0.5
Engines				0.2				0.7											
Equipment_and_furnishings	0.2		0.2			0.2	0.2			0.2									
Fire_protection							0.2			0.2		0.2							
Main_rotor_blades				0.2												0.2			
Main_rotor_head				0.2					0.2						0.2			0.2	0.2
Tail_rotor				0.2					0.2						0.2			0.2	
Transmission			0.2						0.2			0.5				0.2	0.2		0.2
Weapons_and_defence												0.2	0.2						

Note: The values in the matrix are obtained from Cambridge Advanced Modeler (Wynn *et al.*, 2010), which is an implemented toolbox of CPM

Source: Adopted from Clarkson *et al.* (2004)

Figure 5.
DSM of EH101

possible combination might be intractable. The Noisy-OR model can then be used to reduce the knowledge elicitation burden because it requires only the pairwise relationship between each combination of parent and component. In our case example, only the Noisy-OR model is applicable because DSM only identifies the interactions between two components.

One problem of DSM-based information is that the data are intrinsically static. The data cannot be adjusted during the design process once it is obtained. BN, on the other hand, provides intuitive mechanism for updating or learning the network using the engineering change log database. For adjustment of parameters, analysts express the uncertainty about the parameters in the form of their prior distributions. The uncertain parameters are adjusted as the engineering change data are combined with prior distributions to calculate the posterior distributions according to the Bayes' theorem. Even when the DSM data are not available, BN can also automatically construct the network. For details about the parameter learning and updating procedure, please refer to Lee and Hong (2015).

6.3 Scenario I: minimizing overall change propagation risk

After BN-based model is constructed. It can be used to derive optimal freeze sequence. Depending on the objective of the decision maker, a different freeze order can be obtained. The first scenario concerns a case in which the decision maker wants to obtain a design freeze order that minimizes system-level risk. This scenario can be applied to a case in which the main objective of the design freeze is to stabilize the design as early as possible, thereby facilitating convergence to the final design. System-level risk can be obtained by aggregating the change propagation probabilities of the respective components. In this example, the user-defined evaluation function f calculates the average change propagation probability of each of the 19 components as the result of the design freeze decision.

The average change propagation risk would show a monotonically decreasing pattern as the number of frozen components increases. In this case, the myopic algorithm can search the optimal design freeze order. Therefore, only $190(=19(19+1)/2)$ inferences is required in order to obtain the optimal design freeze sequence.

The freeze sequence obtained by the myopic algorithm is provided on the left side of Table II. The first and second columns represent the freeze order and corresponding components, respectively. The third column represents the average change propagation probability of each of the 19 components given the currently frozen components. To illustrate the effectiveness of our methodology, we compared the performance of our solution with that of an arbitrary sequence. This arbitrary sequence is indicated on the right side of Table II. The risk trajectories of the two freeze sequences are illustrated in Figure 6. Unlike the arbitrary sequence, the myopic algorithm rapidly remove the change propagation risk from the system. The area under risk trajectory in Figure 6 was 0.3082 with the arbitrary sequence and 0.2180 with the optimal sequence, indicating that our sequencing method effects a significant risk mitigation improvement. As a result, our myopic sequencing method mitigates more than 30 percent of change propagation risk.

For interpretation of our result, we plot each component with respect to the degree of incoming/outgoing risk. The incoming risk, which is the indicator of change absorber, is obtained by summing the DSM rows. On the other hand, the outgoing risk, the indicator of change multiplier, is obtained by summing the

Seq.	Myopic algorithm Subsystems	Avg. risk	Arbitrary sequence Subsystems	Avg. risk
1	Engines	0.449844	Air conditioning	0.494207
2	Engine auxiliaries	0.410602	Auxiliary electrics	0.493049
3	Flight control system	0.389146	Hydraulics	0.485151
4	Ice and rain protection	0.366664	Ice and rain protection	0.468637
5	Transmission	0.346614	Avionics	0.460993
6	Weapons and defense	0.324962	Fuselage additional items	0.447768
7	Fuselage additional items	0.30282	Fuel	0.440808
8	Main rotor blades	0.278877	Engine auxiliaries	0.39881
9	Tail rotor	0.258822	Flight control system	0.375276
10	Hydraulics	0.239103	Bare fuselage	0.362122
11	Avionics	0.218099	Cabling and piping	0.356957
12	Bare fuselage	0.195187	Engines	0.264193
13	Air conditioning	0.171562	Equipment and furnishings	0.245958
14	Fuel	0.144983	Fire protection	0.225532
15	Fire protection	0.114941	Main rotor blades	0.190475
16	Equipment and furnishings	0.080211	Main rotor head	0.169274
17	Main rotor head	0.050342	Tail rotor	0.137691
18	Cabling and piping	0.024354	Transmission	0.083015
19	Auxiliary electrics	0	Weapons and defense	0

Table II. Design freeze sequence of scenario I

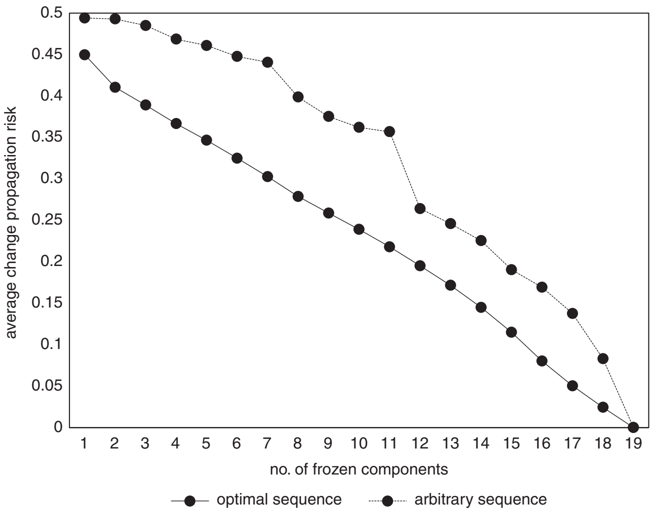
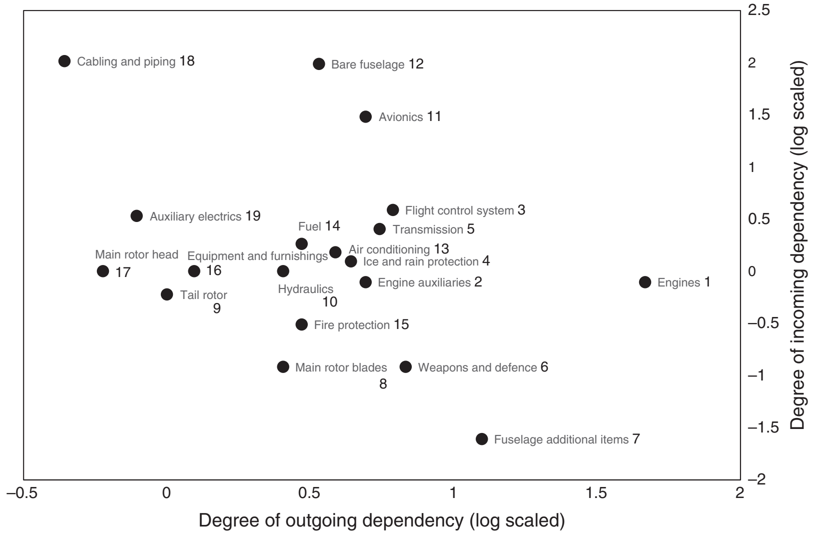


Figure 6. Freeze sequence of scenario I (myopic algorithm vs arbitrary sequencing)

DSM columns. Figure 7 illustrates the relationship between the freeze sequence and the degree of incoming/outgoing risk of each component. As can be seen, the strong change multipliers are likely to be frozen earlier, while the change absorbers are frozen later. However, some exceptional cases can also be found in the figure. For example, Fuselage additional items, which is the second strong multiplier, is frozen in the seventh decision. This may be due to the fact that the relative position on the risk plot continuously changes after each freeze decision is made.



Note: The numbers associated with components indicate optimal freeze sequence obtained from Table II

Figure 7.
Relative position of components

6.4 Scenario II: minimizing change propagation risk of already-frozen components

So far, we have derived a design freeze order that minimizes every component's average change propagation risk that is incurred during the execution of component freezing. However, a decision maker might be interested only in minimizing the risk of already-frozen components. This scenario can be applied to the case in which the redesign cost of an already-frozen component is much more expensive than an unfrozen component. Since our objective is to obtain a freeze sequence that can minimize the change propagation risk of an already-frozen component, now, the evaluation function f computes the average change propagation risk only of frozen components.

The myopic algorithm is first applied to obtain the freeze sequence. The average change propagation risk of frozen components with respect to freeze decisions is illustrated in Table III. Additionally, the performance of the freeze order obtained by arbitrary sequencing is illustrated on the right side of the table. The two different sequencing methods are illustrated in Figure 8. Compared with the first scenarios, the change propagation risk does not show a monotonically decreasing pattern; instead it repeats up and down until it reaches the final components. This is due to the fact that the average risk can increase as the number of frozen components increases. The area under risk trajectory in Figure 8 was 0.3529 with the arbitrary sequence and 0.068 with the myopic algorithm sequence, indicating that our sequencing method provides an almost five times better performance than the arbitrary sequencing method.

Contrary to the first scenario, which shows monotonically decreasing patterns of change propagation risk, in this second scenario, the myopic algorithm cannot guarantee the optimality of the solution. To improve the solution, a non-myopic algorithm can be applied. In this example, specifically, a two-optimal search algorithm is applied. This algorithm first searches component pairs and then rearranges each of them. The results of the two-optimal search algorithm are illustrated in Table IV.

Seq.	Myopic algorithm Subsystems	Avg. risk	Arbitrary sequence Subsystems	Avg. risk
1	Main rotor blades	0.134516	Air conditioning	0.641448
2	Weapons and defense	0.149489	Auxiliary electrics	0.675127
3	Engines	0.148517	Hydraulics	0.554937
4	Engine auxiliaries	0.108071	Ice and rain protection	0.480047
5	Ice and rain protection	0.092182	Avionics	0.533423
6	Tail rotor	0.087999	Fuselage additional items	0.479137
7	Transmission	0.082403	Fuel	0.475601
8	Main rotor head	0.069468	Engine auxiliaries	0.401292
9	Flight control system	0.051856	Flight control system	0.375420
10	Hydraulics	0.041406	Bare fuselage	0.418887
11	Fuselage additional items	0.049941	Cabling and piping	0.460733
12	Avionics	0.057744	Engines	0.327048
13	Fuel	0.061620	Equipment and furnishings	0.300808
14	Fire protection	0.061236	Fire protection	0.265031
15	Air conditioning	0.057300	Main rotor blades	0.214300
16	Equipment and furnishings	0.045304	Main rotor head	0.186738
17	Bare fuselage	0.036918	Tail rotor	0.148091
18	Auxiliary electrics	0.023609	Transmission	0.087152
19	Cabling and piping	0	Weapons and defense	0

Table III. Design freeze sequence of scenario II

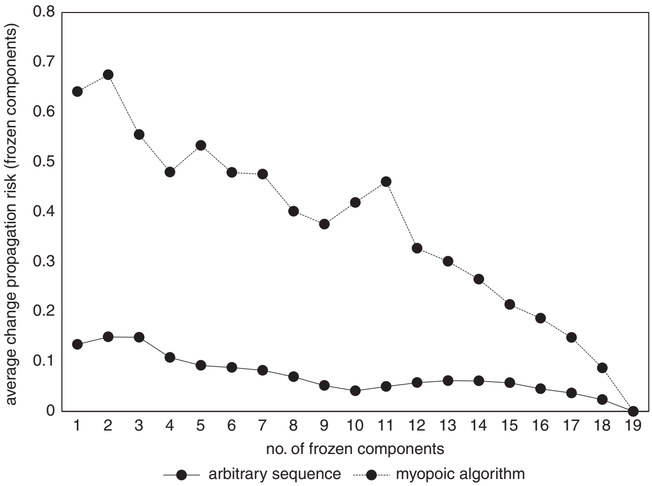


Figure 8. Freeze sequence of scenario II (myopic vs arbitrary sequencing)

As seen in Figure 9, the two algorithms show different freeze orders for the first six components. The area under risk trajectory of the new freeze order was 0.0670, which represents a slight improvement in the performance.

7. Conclusion

Change propagation is the major source of schedule delays and cost overruns in design projects. One way to mitigate the risk of change propagation is to impose a design freeze on components at some point prior to completion of the process.

Table IV. Comparison between myopic algorithm and non-myopic algorithm

	Myopic algorithm		Non-myopic algorithm (2-opt)	
Seq.	Subsystems	Avg. risk	Subsystems	Avg. risk
1	Main rotor blades	0.134516	Engines	0.218953
2	Weapons and defense	0.149489	Engine auxiliaries	0.132653
3	Engines	0.148517	Main rotor blades	0.114560
4	Engine auxiliaries	0.108071	Ice and rain protection	0.098246
5	Ice and rain protection	0.092182	Weapons and defense	0.092182
6	Tail rotor	0.087999	Tail rotor	0.087999
7	Transmission	0.082403	Transmission	0.082403
8	Main rotor head	0.069468	Main rotor head	0.069468
9	Flight control system	0.051856	Flight control system	0.051856
10	Hydraulics	0.041406	Hydraulics	0.041406
11	Fuselage additional items	0.049941	Fuselage additional items	0.049941
12	Avionics	0.057744	Avionics	0.057744
13	Fuel	0.061620	Fuel	0.061620
14	Fire protection	0.061236	Fire protection	0.061620
15	Air conditioning	0.057300	Air conditioning	0.061236
16	Equipment and furnishings	0.045304	Equipment and furnishings	0.057300
17	Bare fuselage	0.036918	Bare fuselage	0.045304
18	Auxiliary electrics	0.023609	Auxiliary electrics	0.036918
19	Cabling and piping	0	Cabling and piping	0.023609

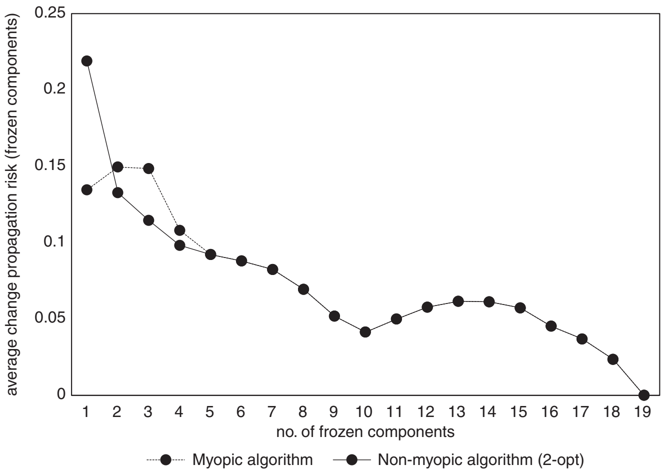


Figure 9. Comparison of myopic and non-myopic algorithms

In this situation, identifying the most appropriate freeze sequence is crucial, as it can effectively mitigate change propagation risk and, thereby, improve design project outcomes.

This paper proposes a BN-based model for deriving the optimal freeze sequence. In this study, DBN was used to represent the change propagation process within a system. According to the model, when a freeze decision is made with respect to a component, a probabilistic inference algorithm within the BN updates the uncertain state of each component. Utilizing this mechanism, we can identify the trajectory of risk according to the freeze sequence. And since identification of the optimal freeze sequence is similar to the sequential decision problem, we propose efficient algorithms for identifying near-optimal solutions. In a case study, we derived the optimal freeze sequence of a helicopter design project from real product development process. The experimental result showed that our proposed method can significantly improve the effectiveness of freeze sequencing compared with arbitrary freeze sequencing.

We believe that our model provides a useful guidance for freezing decisions in complex engineering design projects. Practically, our model can be utilized in following applications:

(1) Planning freeze sequence: using the model, one can derive an optimal freeze sequence plan in advance before the actual design process begins. This might be useful for better planning and structuring the design process.

(2) Assessment of change propagation risk levels: our model can monitor the current level of change propagation risk of each component given the component freeze status. This measure helps a project manager or component designers to check the current design progression and dynamically adjust their change implementation plan.

(3) Justification of freezing proposals: our model can also be used to evaluate the effect of additional freeze decisions given the current freeze status. The project manager can quantitatively evaluate the competing freeze proposals when a multiple of them are proposed. Different types of objective functions can also be used to evaluate each scenario. For example, one can evaluate the freeze proposals by comparing how much change propagation risk can be reduced, or how vulnerable the already frozen components become as a consequence of changes of unfrozen components.

However, it is clear that it does not fully capture current engineering design practices. Some questions and issues deserve further discussion and require future research. One issue is that the sequential freeze process assumed in this paper might be unrealistic in some situations. We assumed that a freeze decision is made one-by-one until every parts within the system is frozen. However, in real engineering design projects, some components, such as outsourced modules, might be uncontrollable; therefore, only a subset of components may be frozen in advance of completion. Moreover, freeze decisions might not be as frequently made as in our model. According to Prasad (1996), a freeze decision is made, at most, twice or three times during engineering design phases. Therefore, in real situations, a chunk of subsystems might be frozen concurrently, and the freeze period might be longer than in current models. Fortunately nonetheless, the BN provides a flexible model for addressing all of the aforementioned issues. To that end, more realistic case studies of real engineering projects remain as future work.

Another issue is that there are be several factors that can affect the design freeze decision. The current model considers only the risk of incurring redesign costs for the components. However, one of the important motivations of design freeze is the reduction of the overall development schedule. For example, Eastman (1980) states that a design project schedule can be reduced by by early freeze of long-lead-time items. Combining our methods with existing project management methods might be an interesting avenue of future research.

References

Bhattacharya, S., Krishnan, V. and Mahajan, V. (1998), "Managing new product definition in highly dynamic environments", *Management Science*, Vol. 44 Nos 11-2, pp. 50-64.

Bobbio, A., Portinale, L., Minichino, M. and Ciancamerla, E. (2001), "Improving the analysis of dependable systems by mapping fault trees into Bayesian networks", *Reliability Engineering & System Safety*, Vol. 71 No. 3, pp. 249-260.

Clarkson, P.J., Simons, C. and Eckert, C. (2004), "Predicting change propagation in complex design", *Journal of Mechanical Design*, Vol. 126 No. 5, pp. 788-797.

Cohen, T., Navathe, S. and Futon, R.E. (2000), "C-FAR, change favorable representation", *Computer-Aided Design*, Vol. 32 No. 5, pp. 321-338.

Cowell, R.G., Verrall, R.J. and Yoon, Y.K. (2007), "Modeling operational risk with Bayesian networks", *Journal of Risk and Insurance*, Vol. 74 No. 4, pp. 795-827.

Eastman, R.M. (1980), "Engineering information release prior to final design freeze", *IEEE Transactions on Engineering Management*, Vol. 27 No. 2, pp. 37-42.

Eger, T., Eckert, C.M. and Clarkson, P.J. (2005), "The role of design freeze in product development", *15th International Conference on Engineering Design (ICED'05)*, *Melbourne, August 15-18*, pp. 1-11.

Eppinger, S.D., Whitney, D.E., Smith, R.P. and Gebala, D.A. (1994), "A model-based methods for organizing tasks in product development", *Research in Engineering Design*, Vol. 6 No. 1, pp. 1-13.

Giffin, M., de Weck, O.L., Bounova, G., Keller, R., Eckert, C. and Clarkson, P.J. (2009), "Change propagation analysis in complex technical systems", *Journal of Mechanical Design*, Vol. 131 No. 8, 081001.

Hamraz, B., Caldwell, N.H. and Clarkson, P.J. (2013), "A holistic categorization framework for literature on engineering change management", *Systems Engineering*, Vol. 16 No. 4, pp. 473-505.

Heckerman, D.E. and Nathwani, B.N. (1992), "An evaluation of the diagnostic accuracy of Pathfinder", *Computers and Biomedical Research*, Vol. 25 No. 1, pp. 56-74.

Heckerman, D., Breese, J.S. and Rommelse, K. (1995a), "Decision-theoretic troubleshooting", *Communications of the ACM*, Vol. 38 No. 3, pp. 49-57.

Heckerman, D., Mamdani, A. and Wellman, M.P. (1995b), "Real-world applications of Bayesian networks", *Communications of the ACM*, Vol. 38 No. 3, pp. 24-26.

Herrmann, J.W. (2010), "Progressive design processes and bounded rational designers", *Journal of Mechanical Design*, Vol. 132 No. 8, 081005.

Huang, Y., McMurran, R., Dhadyalla, G. and Jones, R.P. (2008), "Probability based vehicle fault diagnosis: Bayesian network method", *Journal of Intelligent Manufacturing*, Vol. 19 No. 3, pp. 301-311.

Huchzermeier, A. and Loch, C.H. (2001), "Project management under risk: using the real options approach to evaluate flexibility in R&D", *Management Science*, Vol. 47 No. 1, pp. 85-101.

Jarratt, T.A.W., Eckert, C.M., Caldwell, N.H.M. and Clarkson, P.J. (2011), "Engineering change: an overview and perspective on the literature", *Research in Engineering Design*, Vol. 22 No. 2, pp. 103-124.

Johnson, D.S. and McGeoch, L.A. (1997), "The traveling salesman problem: a case study in local optimization", in Aarts, E.H.L. and Lenstra, J.K. (Eds), *Local Search in Combinatorial Optimization*, John Wiley and Sons, London, pp. 215-310.

Keller, R., Clarkson, P.J. and Eckert, C.M. (2008), "Determining component freeze order: a redesign cost perspective using simulated annealing", *ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (IDETC/CIE 2008), Brooklyn, NY, August 3-6*, pp. 1-10.

Keller, R., Eger, T., Eckert, C.M. and Clarkson, P.J. (2005), "Visualising change propagation", *15th International Conference on Engineering Design (ICED'05), Melbourne, August 15-18*, pp. 280-291.

Kjaerulff, U.B. and Madsen, A.L. (2008), *Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis*, Springer, New York, NY.

Koh, E.C.Y., Caldwell, N.H.M. and Clarkson, P.J. (2012), "A method to assess the effects of engineering change propagation", *Research in Engineering Design*, Vol. 23 No. 4, pp. 329-351.

Krishnan, V., Eppinger, S.D. and Whitney, D.E. (1997), "A model-based framework to overlap product development activities", *Management Science*, Vol. 43 No. 4, pp. 437-451.

Lee, J.H. and Hong, Y.S. (2015), "A Bayesian network approach to improve change propagation analysis", *Proceedings of 20th International Conference on Engineering Design (ICED15), Milan, July 27-30*.

Li, Z., Harman, M. and Hierons, R.M. (2007), "Search algorithm for regression test case prioritization", *IEEE Transactions on Software Engineering*, Vol. 33 No. 4, pp. 225-237.

Maier, J.F., Wynn, D.C., Biedermann, W., Lindemann, U. and Clarkson, P.J. (2014), "Simulating progressive iteration, rework and change propagation to prioritise design tasks", *Research in Engineering Design*, Vol. 25 No. 4, pp. 283-307.

Matthews, P.C. (2011), "Challenges to Bayesian decision support using morphological matrices for design: empirical evidence", *Research in Engineering Design*, Vol. 22 No. 1, pp. 29-42.

Matthieu, G., François, P. and Tchangani, A. (2012), "Optimising end-of-life system dismantling strategy", *International Journal of Production Research*, Vol. 50 No. 14, pp. 3738-3754.

Mirarab, S. and Tahvildari, L. (2007), "A prioritization approach for software test cases based on Bayesian networks", in Dwyer, M.B. and Lopes, A. (Eds), *Fundamental Approaches to Software Engineering*, Springer, Berlin Heidelberg, pp. 276-290.

Mookerjee, V.S. and Mannino, M.V. (1997), "Sequential decision models for expert system optimization", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 9 No. 5, pp. 675-687.

Morkos, B., Mathieson, J. and Summers, J.D. (2014), "Comparative analysis of requirements change prediction models: manual, linguistic, and neural network", *Research in Engineering Design*, Vol. 25 No. 2, pp. 139-156.

Moullec, M.L., Bouissou, M., Jankovic, M., Bocquet, J.C., Réquillard, F., Maas, O. and Forgeot, O. (2013), "Toward system architecture generation and performances assessment under uncertainty using Bayesian networks", *Journal of Mechanical Design*, Vol. 135 No. 4, 041002.

Mussi, S. (2002), "Sequential decision-theoretic models and expert systems", *Expert Systems*, Vol. 19 No. 2, pp. 99-108.

Oh, S., Park, B., Park, S. and Hong, Y.S. (2007), "Design of change-absorbing system architecture for the design of robust products and services", in Jacko, J.A. (Ed.), *Human-Computer Interaction. HCI Applications and Services*, Springer, Berlin Heidelberg, pp. 1110-1119.

Prasad, B. (1996), *Concurrent Engineering Fundamentals: Integrated Product and Process Organization*, Vol. I, Prentice Hall, Upper Saddle River, NJ.

Shahan, D.W. and Seepersad, C.C. (2012), "Bayesian network classifiers for set-based collaborative design", *Journal of Mechanical Design*, Vol. 134 No. 7, 071001.

Skaanning, C., Jensen, F.V. and Kjærulff, U. (2000), "Printer troubleshooting using Bayesian networks", in Palm, G. and Ali, M. (Eds), *Intelligent Problem Solving. Methodologies and Approaches*, Springer, Berlin Heidelberg, pp. 367-380.

Thomke, S.H. (1997), "The role of flexibility in the development of new products: an empirical study", *Research Policy*, Vol. 26 No. 1, pp. 105-119.

Vomlel, J. (2004), "Bayesian networks in educational testing", *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 12 No. 1, pp. 83-100.

Wynn, D.C., Caldwell, N.H. and Clarkson, P.J. (2010), "Can change prediction help prioritise redesign work in future engineering systems?", *Proceedings of DESIGN 2010, the 11th International Design Conference*, *Dubrovnik, May 17*.

Yang, F. and Duan, G.J. (2012), "Developing a parameter linkage-based method for searching change propagation paths", *Research in Engineering Design*, Vol. 23 No. 4, pp. 353-372.

Zirger, B.J. and Hartley, J.L. (1996), "The effect of acceleration techniques on product development time", *IEEE Transactions on Engineering Management*, Vol. 43 No. 2, pp. 143-152.

Further reading

Cheng, H. and Chu, X. (2012), "A network-based assessment approach for change impacts on complex product", *Journal of Intelligent Manufacturing*, Vol. 23 No. 4, pp. 1419-1431.

Huang, G.Q. and Mak, K.L. (1999), "Current practice of engineering change management in UK manufacturing industries", *International Journal of Operations & Production Management*, Vol. 19 No. 1, pp. 21-37.

Jensen, F.V. (1996), *An Introduction to Bayesian Networks*, Springer, New York, NY.

Kang, C.M. and Hong, Y.S. (2009), "Evaluation of acceleration effect of dynamic sequencing of design process in a multiproject environment", *Journal of Mechanical Design*, Vol. 131 No. 2, 021008.

Pearl, J. (1988), *Probabilistic Reasoning in Intelligent Systems: Network of Plausible Inference*, Morgan Kaufmann Publishers, Burlington.

About the authors

Jihwan Lee is a PhD Candidate of Industrial Engineering at the Seoul National University, Korea. He received the BS degree in industrial engineering from Seoul National University. His research interests are in the area of product and service development, product architecture, and platform design.

Yoo S. Hong is an Associate Professor of Industrial Engineering at the Seoul National University, Korea. He received his BS and MS degree from the Seoul National University and earned his PhD from the Purdue University, USA. He taught at the University of Toledo, USA, as an Assistant Professor before he took the current position. His current research interest lies in the area of product and service engineering. The topics of major interest include product architecture and platform design, new product/service development, sustainable product design, product life-cycle management, and product-development processes. He is a member of KIIE, ASME, INFORMS, and Design Society. Professor Yoo S. Hong is the corresponding author and can be contacted at: yhong@snu.ac.kr

This article has been cited by:

1. David C. Wynn, Claudia M. Eckert. 2016. Perspectives on iteration in design and development. *Research in Engineering Design* . [CrossRef]