Standard Article

Concurrent Engineering: Research
and Applications
1–15

DOI: 10.1177/1063293X17708820
journals.sagepub.com/home/cer
SAGE

Multi-variation propagation prediction based on multi-agent system for complex mechanical product design

Songhua Ma, Zhaoliang Jiang and Wenping Liu

Abstract

The design change propagation in a complex product is characteristic of nonlinear, dynamic, and uncertain; its impact analysis becomes challenging. Furthermore, the multiple changes occurring concurrently during the whole design process increase the difficulty of the change propagation analysis. In this article, a new change propagation prediction method based on multi-agent system, especially for multi-variations, is proposed to improve the validity of the change routing paths. As the foundation of the proposed prediction method, a hierarchical network composited of the specific design properties is used to represent the design change analysis model. Based on the discussion of concurrent propagation patterns in the context of multi-variations, the quantitative definition of change propagation impact focuses on two factors: change impact and propagation likelihood. To reduce the subjective measurements estimated through the designer's experience, the change impact and the propagation likelihood are quantified by evaluating the design specification and mining the design change records, respectively. Based on the quantified change propagation impact, the multi-variation propagation simulation is parameterized and implemented in the platform of NetLogo. To demonstrate the effectiveness of the proposed method, experiments are conducted with one typical scenario. Compared with the classical change propagation methods, the proposed method proves to be valid for analyzing the concurrent change propagation, and the analysis reliability is guaranteed.

Keywords

multi-variation propagation, design change analysis model, multi-agent system, concurrent propagation, change propagation prediction

Introduction

The development of a new mechanical product can be interrupted by a wide range of design changes from the design requirements, technological innovation, political environments, and so on. With very different causes, design changes can be classified into two main categories: *emergent design change* and *initiated design change*. The former is caused by the problems occurring across the internal design project due to solution uncertainty, whereas the latter is caused from external stakeholders such as new customer requirements, technological innovations, and regulation modifications (Eckert et al., 2004). All these changes have a significant impact on the outcome of product development. According to the statistics, the design changes determine as much as 70%–80% of the final cost of a product (McIntosh, 1995) and bring uncertainties to the development schedule and quality. By analyzing the change records of certain original equipment manufacturers, Shankar et al. (2012) found that 77.0% of changes were derived from internal reasons, while 23.0% were external and inferred that 32.4% of the total changes resulted from the change propagation.

Change propagation is a process in which a change to one element of an existing design tends to trigger additional changes to other elements of the same design

Key Laboratory of High Efficiency and Clean Mechanical Manufacture, Ministry of Education, Shandong University, Jinan, P.R. China

Corresponding author:
Songhua Ma, Key Laboratory of High Efficiency and Clean Mechanical Manufacture, Ministry of Education, Shandong University, Jinan 250100, P.R. China.
Email: msh_1216@sdu.edu.cn

in a cause-effect-cause-effect pattern. This propagation will not be completed until the design achieves a new stable status. In other words, such loop-like, dynamic, and recursive change propagation processes continue until all of the inconsistencies are identified. Consequently, it is even possible for changes that were initially thought as simple to propagate uncontrollably, resulting in an avalanche propagation. This is especially true for the design of complex products composed of tightly coupled elements and functionalities. Considering the fact that certain design properties are tightly coupled and numerous branches should be routed, the change propagation prediction can become more challenging even if the product under design undergoes minor changes. Numerous random or uncertain changes that often conflict with each other may occur in a design process and increase the risk of product development. Therefore, it is significant to predict the change propagation and route the change to satisfy the design requirement with low cost. Predicting the impact of the changes helps the design reviewers identify and evaluate the effects of the change.

Multi-variations are short for the multiple changes being concurrently triggered on different elements or systems in the complex mechanical product design. During the concurrent propagation processes, they often reduce or amplify each other at the coupled node. Some changes may increase the value of the coupled node, whereas the others may decrease it. This kind of change conflict is common during the concurrent design process and even causes repetitive redesign work. Thus, it is urgent to propose a technique to predict how multi-variations affect the rest of product and to evaluate which nodes need to be modified accordingly.

To assess the impact caused by the multi-variations, a change propagation prediction approach based on the multi-agent technology is proposed for the complex mechanical product design. The design properties and the specifications of the development process can be represented with a hierarchical network model, and the system-level impact resulting from the multi-variations can be assessed quantitatively.

Related work

Engineering change research can be traced back to the early 1980s when the first publication was contributed by Diprima (1982). Due to the limitations of less-advanced technologies, previous research focused on the consistency of information during changing. This section introduces the state-of-the-art change propagation prediction approaches from two aspects: analysis model and prediction method.

Design change analysis model

The design attributes, functions, and requirements can all be represented as components connected via inter-component links in the design change analysis model (DCAM). Prasad (2002) defined four kinds of design variables that would be subject to change including size variable, shape variable, topology variable, and process variable to capture a system-level optimization as part of a product design process. The most famous study on DCAM is the contribution from Clarkson et al. (2004), who built a design structure matrix (DSM) as DCAM, according to the parameter relationships of the design components. Some later studies modified the DSM to exclude the propagation loop path and self-dependent path (Hamraz et al., 2013a) and to analyze the requirement change propagation (Morkos et al., 2012). Based on the DSM, Li and Chen (2014) utilized a design dependency matrix (DDM) to organize the dependencies between design parameters and functions. Except for DSM, Cohen et al. (2000) proposed a C-FAR (Change Favorable Representation) matrix to describe the product in the form of attribute/value to analyze the change impact. Based on the house of quality (HoQ), Koh et al. (2012) modeled the effects of the potential change propagation generated by product components, change options, and product requirements. After the function-behavior-structure (FBS) ontology was proposed by Gero (1990), its modifications were introduced for modeling DCAM (Ahmad et al., 2013; Fei et al., 2011). As an extension of FBS, Pasqual and De Weck (2012) introduced a multilayer network model including three coupled layers, that is, product layer, change layer, and social layer.

To simulate the change impact, the design activities or tasks were often defined as the components of the DCAM in several studies. In particular, the dependencies of design activities were represented as DSM to evaluate the change impact in Chua and Hossain's work (Chua and Hossain, 2012). Seeing works from Wynn et al. (2014) and Li et al. (2012), the design tasks were modeled as the nodes of a complex design workflow to predict change propagation and to select the most economic propagation path. Li and Moon (2012) organized the design activities as a process to investigate how the requirements and technology changes eventually affect the leading time, cost, and quality.

Recently, the network-based DCAM has attracted the attention of researchers. Cheng and Chu (2012) considered the complex product as a weighted network composed of parts, subassemblies, and subsystems. To analyze the variation propagation of the quality characteristic (QC), Duan and Wang (2013) built a QC-linkage network, which was constructed by the design properties, parameter relationships, and constraint relationships. Lee et al. (2010) introduced the analytic network process (ANP)-based approach to measure the relative weight of parts and modules in a modular product in terms of design change impacts. Reddi and Moon (2011) developed a dynamic system model for a collaborative supply chain to study effective and efficient engineering change management.

Change propagation prediction method

At the macro level, the effects of change propagate in three patterns: (1) *ripple*, which triggers a small and fast-decreasing volume of changes; (2) *blossom*—a large number of changes which oscillate and turn to be convergent within expected limits; and (3) *avalanche*—an increasing volume of changes that may not be brought to a conclusion after a given end point (within a certain time or number of changes). The quantification of the measurements is essential to the validity of the change propagation prediction. In previous research, design experts were responsible for estimating the measurements of the change propagation effort, such as propagation likelihood and change impact in most DSM-based research (Clarkson et al., 2004). However, the measurements extracted from the existing change records and the characteristics of DCAM are more objective. Duan and Wang (2013) adopted several variation mitigation methods to reduce the change impact, such as source uncoupling, variation compensation, variation deployment, linkage sensitiveness, linkage principle, superposing effect variation, and propagation path variation. Ouertani (2008) evaluated the change uncertainty conditions using the variability, sensitivity, and completeness of the nodes in a data dependency network. Mehta et al. (2013) quantified the important attribute sets by the information entropy to capture the knowledge from the existing engineering changes. Tang et al. (2016) found the optimal solution of the design change propagation by examining the workload of each change propagation path.

The change propagation impact can be simply evaluated by k copies of the adjacency matrix extracted from DCAM, where k corresponds to the number of walks from the initial changed element to the others. However, searching change propagation path can be abstracted into a traveling salesman problem known to be NP-hard (non-deterministic polynomial-time hard). Yang and Duan (2012) filtered the optimal change propagation paths based on certain selection strategies. Li and Zhao (2014) applied the genetic algorithm to find the optimal propagation likelihood for each optional propagation path. Furthermore, they modified the breadth-first search method to locate the shortest paths of change propagation (Li et al., 2016). Ni Li et al. (2015) utilized multi-agent systems (MASs) to assess the risk propagation for complex product design.

Current issues

Most of the previous research works modeled the product as a network of elements (i.e. systems, components, or parts) that were linked by their dependencies (i.e. structural, behavioral, and functional parameters) to construct an available DCAM and further described the change propagation as the spread of knock-on effects along the links of this network. Although the previous methods are helpful tools in analyzing change propagation, there are three primary issues that prevent the current change propagation prediction method being applied:

1. In most of the previous research works, each component in DCAM corresponds to a design part or a subsystem (Clarkson et al., 2004). The rough organization prevents the valid impact assessment. One reason for this may be due to the subjective measurements evaluated from the design experience, which often deviate from the actual values. In this case, it is important to propose a quantitative metric and an objective measuring method. Building the DCAM at the property/parameter level would help assess the impact of change propagation objectively and quantitatively. Obviously, the change prefers the path with the larger propagation likelihood and the lower change impact to propagate to guarantee that the change propagation converges rapidly.
2. Chua and Hossain (2012) found that five factors—the transition matrix, degree of initiated change, timing of initiated change, point of initiated change, and redesign duration—affect the change propagation. The previous research neglected to distinguish the different change propagation impacts resulting from the different volumes of the initiated change. Consequently, even if the volume of initiated change varies, the results from the previous prediction method remain the same since the DCAM and the initial change unvaried. It is essential to predict the change propagation paths that can vary with the variation of the initiated changes.
3. In complex product design, in particular, several changes may be triggered on different elements or systems at the same time, which often interact with each other. It is necessary to negotiate about the volume of variation to resolve the conflict from multi-variations. Furthermore, multi-variations would generate the propagation loop sometimes and cause repeated redesign work that is unfavorable in the design process. The propagation loop

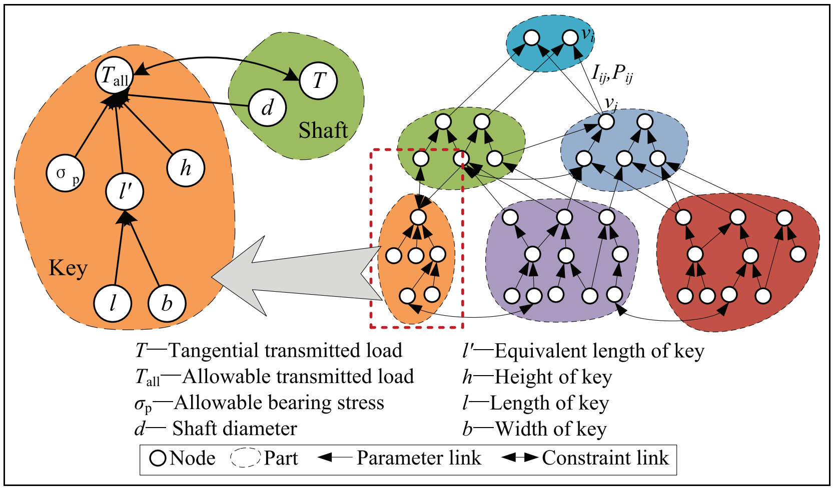


Figure 1. A design property network—DCAM.

should be predicted and resolved before the change being implemented. The previous research focuses on studying the propagation impact delivering from a sole initiated change (Yang and Duan, 2012). However, the multi-variations occurring and propagating concurrently in the complex design processes are more complicated to analyze, which are the causes for concern.

Change propagation foundation—DCAM

Design changes are changes and/or modifications to the released structure (fits, forms and dimensions, surfaces, materials, etc.), behavior (stability, strength, corrosion, etc.), function (speed, performance, efficiency, etc.), or the relations between function and behavior (design principles), or behavior and structure (physical laws) of a technical artifact (Hamraz et al., 2013b). The structural attributes, behavioral attributes, and functional attributes are unified as design property. This article focuses on analyzing the change impact resulting from the variations of some design properties. In other words, the parameter value of some design properties varies with the change propagation, while the relations between design properties and the topology of the DCAM remain the same.

As a foundation to the multi-variation impact analysis, DCAM is defined in the form of $G = \{V, C, E\}$, where $V = \{v_1, v_2, \ldots, v_N\}$ is the node set, $C = \{c_1(t), c_2(t), \ldots, c_N(t)\}$ is the value set, and $E = \{e_{ij} \in V \times V, i, j = 1, 2, \ldots, N; i \neq j\}$ is the link set as shown in Figure 1. The nodes are represented with circles in the DCAM diagram. Each node corresponds to a design property. Different design properties are interdependent according to their geometric dimensions, function, behavior, material, and other specifications. They are connected by links with directed arrow. $c_i(t)$ is the current value of node v_i in the tth propagation step. The arrow of link points from child nodes to parent nodes as the direction of the design specification flow. One link represents a function in the form of $v_j = f(v_0, v_1, \ldots, v_i)$, where v_j is the parent node and each v_i is a child node. Based on these definitions, DCAM is capable of demonstrating how design properties in the bottom layer determine properties in the top layer. In Figure 1, the enlarged part of the schematic DCAM shows the design specification shearing resistance design of key. Additionally, two functions, that is, $f_1(h, l', d, \sigma_p) = T_{all}$ and $f_2(l, b) = l'$ are represented by links.

The change to a design property cannot be implemented directly. It should be translated into the change of its connecting properties. Since the parent and child nodes are connected by link, the value of the parent node varies with the changes delivered from its child nodes according to the physical laws and the design specification, that is

$$\Delta v_j = \frac{\partial f(v_0, v_1, \ldots, v_i)}{\partial v_i} \Delta v_i = I_{i,j} \Delta v_i \qquad (1)$$

where $I_{i,j}$ is defined to quantify the change impact. $I_{i,j}$ measures the proportion of the variation of the successor node due to the variation from its predecessor nodes, which can be evaluated through the derivation of the design function about v_i. Additionally, there is another kind of link with double arrows in the DCAM, which represents the artificial rule in the form of $f(v_{i-1}, v_i, \ldots, v_k) \geq 0$. For example, $f_3(T_{all}, T) \geq 0$. The inequality can be transformed into the general formula $f(v_{i-1}, v_i, \ldots, v_k) = \Delta v_j$ by adding the difference node

Δv_j as referring to Yang and Duan's (2012) work. After recursively defining nodes and searching for their child nodes, the DCAM is then constructed as a complex network composed of nodes and links. Consequently, the DCAM shows how the design properties in the bottom layer determine others in the top layer.

A design property can be both the child node in one link and the parent node in another, such as the property l' in the aforementioned example. The nodes connected with only their parent nodes or child nodes form the boundary of DCAM. After an initial change is triggered, its propagation continues until reaching the boundary nodes. The variation from the lower layer node can propagate upward to the upper layer nodes and even cause the impact diffusion. Conversely, the upper layer variation should be decomposed and transmitted to its child nodes, until the boundary nodes are fixed.

Change propagation pattern

Based on the constructed DCAM, the changes of some design properties can propagate along the links. To evaluate the variations of the impacted nodes, the change propagation pattern in the DCAM is concluded in this section, which can be categorized into three: single, multiple, and loop-like.

Sole change propagation pattern

Once an initial change on a certain node is triggered, it will invoke any of the connecting nodes to change. Then, the change propagation starts. The next propagation step would act as the previous step recursively. After studying two-step propagation, the change propagation can be divided into three patterns: serial, parallel, and hybrid. In the serial propagation pattern, the change consistently chooses the parent nodes (child nodes) in each step as shown in Figure 2(a) and (b). If the change propagates along the direction of the link, it is named as downstream propagation pattern; on the contrary, it is named as upstream propagation pattern. The downstream propagation makes the parent node adapt with the variation from the lower layer, whereas the upstream propagation drives the variation of the child node to satisfy the change requirement from the upper layer. The serial upstream propagation ends up with the root node and varies the design requirement ultimately, whereas the serial downstream propagation ends up with the leaf node that is permitted to change. In the parallel propagation pattern, the change concurrently chooses its several parent nodes (child nodes) to propagate as shown in Figure 2(c) and (d). Since this propagation pattern makes the change diffuse, the propagation path should be routed to decrease the impact from the diffusion. Likewise, the parallel propagation pattern can be divided into the upstream pattern and the downstream pattern according to the difference between the propagation direction and the link direction. The parallel downstream propagation will generate more design risks than the upstream does since it will impact more upper layer nodes. The third propagation pattern is a hybrid of the upstream and downstream patterns, which tends to choose the sibling nodes to propagate as shown in Figure 2(e) and (f). In Figure 2(e), the intermediate node v_{11} may even not be impacted by the variation from node v_{22} since the variation will be diverted to node v_{21} and node v_{23}. Generally, the actual change propagation is a combination of these patterns.

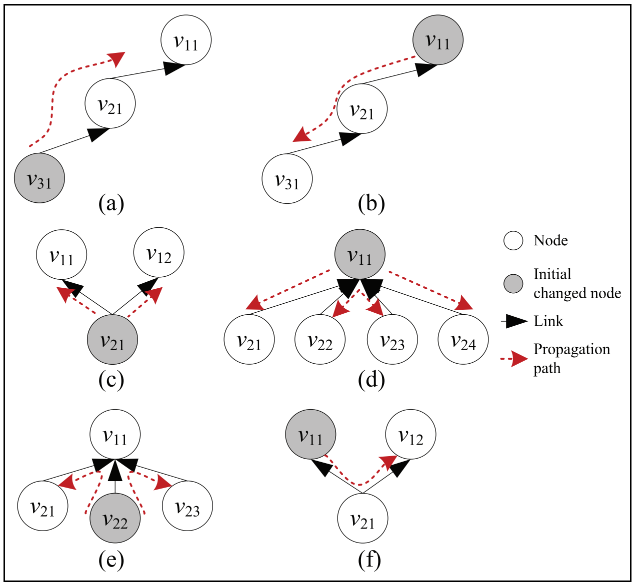


Figure 2. Single change propagation pattern in DCAM (a) Downstream serial; (b) Upstream serial; (c) Downstream parallel; (d) Upstream parallel; (e) Downstream hybrid; (f) Upstream hybrid.

Coupled change propagation pattern

During the concurrent propagation processes, the change requirements converging on the coupled node may be different, which causes the change conflict and the repetitive redesign work. According to the different change propagation patterns converging on the coupled node, three categories of conflicts between two changes are sketched in Figures 3 to 5, respectively. In Figure 3(a), two downstream change propagations converge on the coupled node, which enlarge their shared parent node v_{21} in a different volume. The change requirement from node v_{11} is more than that from node v_{12}. Then, the value of node v_{21} can be changed into the summation of the variations of node v_{11} and node v_{12}, that is

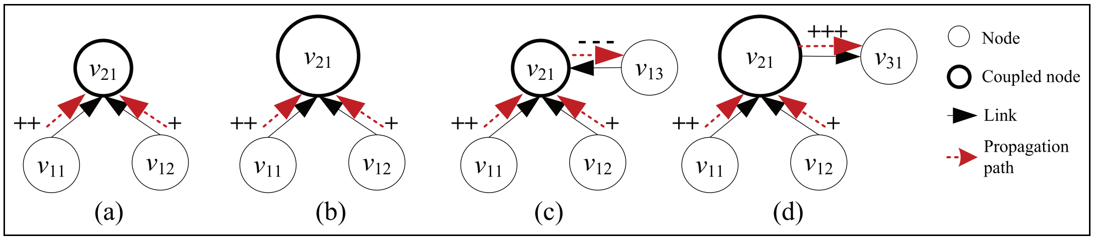


Figure 3. Two downstream change propagations converging on a coupled node (a) Propagations coupled; (b) Without other node; (c) With child node; (d) With parent node.

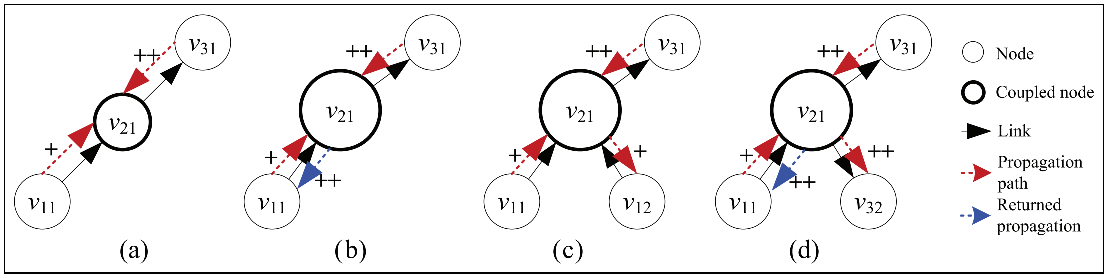


Figure 4. One downstream propagation and one upstream propagation converging on a coupled node (a) Propagations coupled; (b) Without other node; (c) With child node; (d) With parent node.

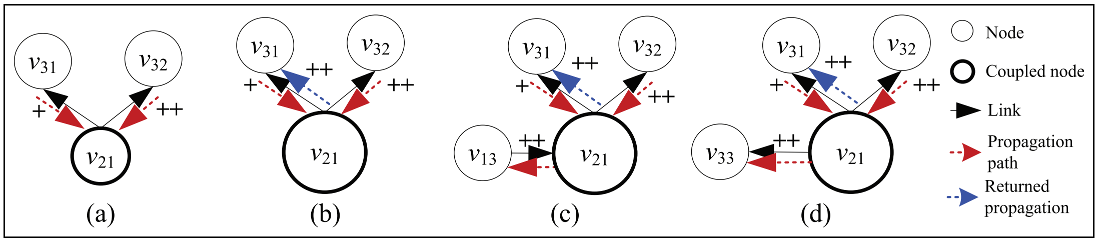


Figure 5. Two upstream change propagations converging on a coupled node (a) Propagations coupled; (b) Without other node; (c) With child node; (d) With parent node.

$$
\begin{aligned}
\Delta v_{21} &= f(v_{11}+\Delta v_{11}, v_{12}+\Delta v_{12}) - f(v_{11}, v_{12})\\
&= f(v_{11}, v_{12}) + \sum_{i=1}^{\infty}\left(\Delta v_{11}\frac{\partial}{\partial v_{11}} + \Delta v_{12}\frac{\partial}{\partial v_{12}}\right)^{i}\\
&\quad f(v_{11}, v_{12}) - f(v_{11}, v_{12}) \qquad (2)\\
&\approx \Delta v_{11}\frac{\partial f(v_{11}, v_{12})}{\partial v_{11}} + \Delta v_{12}\frac{\partial f(v_{11}, v_{12})}{\partial v_{12}}\\
&= \Delta v_{11} I_{11,21} + \Delta v_{11} I_{12,21}
\end{aligned}
$$

According to the character of the other nodes connecting v_{21}, the next step of change propagation presents three patterns. In Figure 3(b), since node v_{21} does not connect with other child nodes except node v_{11} and v_{12}, its value is enlarged into $c_{21} + \Delta v_{21}$ to satisfy the design specification and the change propagation ends. Conversely, if v_{21} has the other child nodes such as v_{13}, the change will be transmitted to the child nodes; the variation Δv_{21} can be decreased by varying v_{13} as shown in Figure 3(c). The variation Δv_{21} is unfavorable if v_{21} has other parent nodes such as v_{31} as shown in Figure 3(d). In this case, it may generate a further change diffusion to the nodes in the upper layer with a great volume.

In Figure 4(a), two opposite change propagations converge on the coupled node. One is downstream, and the other is upstream. The upstream propagation requires node v_{21} to be changed into $c_{21} + \Delta v_{21}$, whereas the downstream propagation v_{11} can only provide variation $\Delta v'_{21}$ to node v_{21}. Suppose $\Delta v'_{21} < \Delta v_{21}$, the change conflict between the two opposite change propagations can be written as

$$
\begin{aligned}
\Delta v_{21} - \Delta v'_{21} &\approx \frac{\Delta v_{31}}{\left(\frac{\partial f(v_{21}, v_{31})}{\partial v_{21}}\right)} - \Delta v_{11}\frac{\partial f(v_{11}, v_{21})}{\partial v_{11}} \qquad (3)\\
&= \frac{\Delta v_{31}}{I_{21,31}} - \Delta v_{11} I_{11,21} = \Delta v_{31} I_{31,21} - \Delta v_{11} I_{11,21}
\end{aligned}
$$

To prevent further change diffusion to the upper layer nodes, node v_{21} is enlarged into Δv_{21} as the change requirement from node v_{31}. Furthermore, the variation of v_{21} returns to node v_{11} to form a new upstream propagation according to the design specification as indicated by the blue arrow in Figure 4(b). In Figure 4(c), since node v_{21} has another child node v_{12} besides v_{11}, the conflicting variation $\Delta v_{21} - \Delta v'_{21}$ can be routed to node v_{12} to satisfy the change requirements from both v_{11} and v_{31} at the same time. If v_{21} has another parent node v_{32}, its variation Δv_{21} will certainly diffuse to v_{32} as shown in Figure 4(d).

In Figure 5(a), two upstream change propagations converge on a coupled node. The change requirements from parent node v_{31} and node v_{32} are different, where suppose $\Delta v_{21} < \Delta v'_{21}$. The change conflict between Δv_{21} and $\Delta v'_{21}$ can be expressed as

$$\begin{aligned}\Delta v'_{21} - \Delta v_{21} &\approx \frac{\Delta v_{32}}{\frac{\partial f(v_{21}, v_{32})}{\partial f(v_{21})}} - \frac{\Delta v_{31}}{\frac{\partial f(v_{21}, v_{31})}{\partial f(v_{21})}} \\ &= \frac{\Delta v_{32}}{I_{21,32}} - \frac{\Delta v_{31}}{I_{21,31}} = \Delta v_{32} I_{32,21} - \Delta v_{31} I_{31,21}\end{aligned} \tag{4}$$

Suppose that node v_{21} is enlarged with $\Delta v'_{21}$ as the change requirement from v_{32}, node v_{31} varies according to the variation of v_{21} to form a new downstream propagation as indicated by the blue arrow in Figure 5(b). This downstream propagation can never be routed no matter whether node v_{21} has another child node as shown in Figure 5(c) or other parent nodes as shown in Figure 5(d).

In fact, the variation could be positive or negative. All the examples use the positive variation to illustrate the change propagation pattern, which does not impact the conclusion. If the multiple changes converge on the coupled node, which can be analyzed as the aforementioned examples, the conclusion remains the same. To the couple node v_i, its variation is determined by summing the variation of its neighboring node v_j as the aforementioned analysis. The temporal value of v_i at the next propagation step $t + 1$ can be calculated as the summation of all the variations from the neighboring nodes at the current step t as follows

$$\begin{aligned}&c_i(t+1) = c_i(t) + \max \\ &\left(\sum_{j \in F_i^{\text{child}}} \big(c_j(t) - c_j(t-1)\big) I_{j,i},\ \max_{j \in F_i^{\text{parent}}} \Big(\big(c_j(t) - c_j(t-1)\big) I_{j,i} \Big) \right)\end{aligned} \tag{5}$$

where F_i^{child} is the set of child nodes of v_i, and F_i^{parent} is the set of parent nodes of v_i.

Loop-like change propagation

It is common to confront different change requirements during the complex mechanical product design. One needs the couple node changed into the variation Δc_i and the other only requires $\Delta c'_i$ ($\Delta c_i \neq \Delta c'_i$). This kind of conflict between multiple change requirements should be resolved by planning the change propagation path. Furthermore, if the conflicting changes propagate along a closed-loop path, which would never be eliminated, they cause the loop-like propagation and the repetitive change. To avoid the loop-like propagation and repetitive changes, the conflicting change requirements on the closed-loop path should make some negotiations about the variations before being implemented. Consequently, the change requirements are not equal to the initial after the trade-off negotiations.

During the negotiation process, the change requirement of node v_i is impacted by that of its neighboring nodes which belong to the closed-loop path, that is

$$\Delta c_i(\tau + 1) = I_{j,i} \Delta c_j(\tau) \tag{6}$$

where τ represents the negotiation step that differs from the definition of propagation step t. In general, the above function can be transformed into matrix form as

$$\Delta \boldsymbol{C}(\tau + 1) = \mathbf{I} \Delta \boldsymbol{C}(\tau) \tag{7}$$

where $\mathbf{I}$ is the impact matrix. The solution is directly determined by the eigenvalues of $\mathbf{I}$. Consequently, the stable change requirements after the negotiation are determined by the eigenvalues of $\mathbf{I}$. Then, the eigenvalues of $\mathbf{I}$ can be utilized to analyze the ability of agreement. The necessary (but not sufficient) condition of reaching the agreement is $\sigma(\mathbf{I}) < 1$, where σ is the maximum non-trivial eigenvalue of $\mathbf{I}$ or named spectral gap. If $\sigma(\mathbf{I}) < 1$, the conflicts between the change requirements of the related nodes would decrease exponentially. The negotiation terminates until the change requirements from all the neighboring nodes do not vary.

Generally, it is not common to satisfy the condition $\sigma(\mathbf{I}) < 1$ to the nodes on the closed-loop path. Taking a closed-loop path containing three nodes, for example, suppose the initial change requirements of the nodes are 0.5, 1.0, and 0.25, respectively, that is, $\Delta c_0(0) = 0.5$, $\Delta c_1(0) = 1.0$, and $\Delta c_2(0) = 0.25$ as shown in Figure 6(a). The change impact between each other can be found in the diagram. In this case, the negotiation of the change requirements can reach an agreement, although $\sigma(\mathbf{I}) = 1$. If $\Delta c_2(0)$ is changed into -0.25 as shown in Figure 6(b), the change requirements of three nodes disagree with each other at the initial negotiation step. Then, the change requirements oscillate on the closed-loop path according to equation (6). In this case, the change conflicts cannot be resolved by the negotiation unless the change requirements jump out of the closed-loop path and search for another propagation path. In Figure 6(c), the impact matrix is changed, and

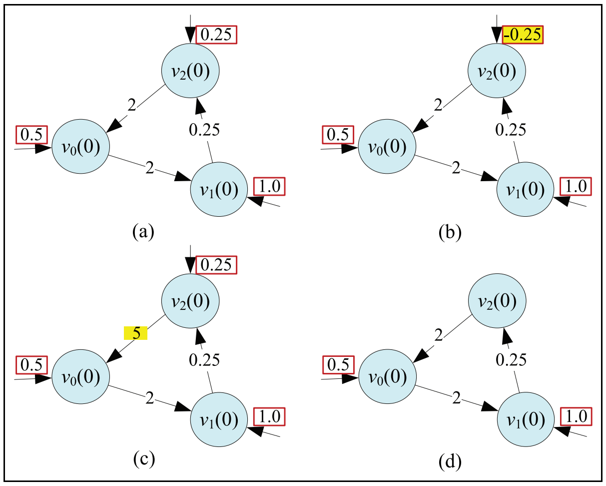


Figure 6. Negotiation process between two conflicting change requirements (a) Initial negotiation; (b) Requirement changed; (c) Impact value changed; (d) Dimension changed.

$\sigma(\mathbf{I}) > 1$. It is not appropriate for the convergence condition. This closed-loop path should be blocked for change propagating, and the change requirements should be routed to other branches. If node v_2 does not receive any change requirement as shown in Figure 6(d), the dimensionality of the impact matrix is reduced to 2×2, which only contains the change impact between node v_0 and v_1. Its analysis is the same as the aforementioned cases.

Change propagation simulations in MAS

For a DCAM with M links, an initial change may lead to propagation in approximate 2^M paths. Since the number of branches dramatically increases with M, it is too complex to obtain the change propagation path by manual search. Furthermore, the multi-variations would interact with each other during the propagation, which brings much more dynamic and uncertain factors to analysis. The realization that the analysis of the multi-variation propagating in the complex mechanical product design process is a nonlinear and complex problem is one of the key motivations for simulating the change propagation using a MAS. Before predicting the multi-variation propagation with the help of MAS, the variables and rule of each agent are defined in this section.

MAS

An MAS consists of multiple intelligent agents interacting with each other within an operating environment. In this article, an agent performs the essential characteristics of autonomous behaviors rather than be a purely passive component. Agents should obey the lower level rules to guide their behaviors and have the higher level permissions to change lower level rules themselves. Through applying MAS, it is possible to represent nonlinear, dynamic, and stochastic characteristics of a complex system through the defined agents and their interactions. MAS has been proved to be valid and been applied widely in previous research, such as transportation systems, social research, infectious disease science, and environment evolution.

In our MAS, the DCAM is replaced by an agent-based interactive network. Each node is represented by an agent, which interacts with others through links. In the MAS, all the agents are connected in the network with a given topological structure. Each agent in the network is subjected to change according to the variation of its neighboring agent and assigned a local bound constraint, which is not known by other agents. The agent exchanges information with its neighbors locally; then, all the agents fulfill the task of simulation cooperatively. The role of an agent is intelligent to vary itself within a constraint under the impact from the neighboring agent. All the agents are responsible for simulating the change propagation process until the propagation reaches the network boundary. Each agent is responsible for adding the variations from its precedent agent(s) to its variable. Along the links, the change flow will continue to next agents.

To simulate the multi-variation propagation processes, another variable of a link is defined as propagation likelihood $P_{i,j}$ besides the change impact $I_{i,j}$, which is defined as the possibility of the change propagating along the link. Prior to the simulation, the propagation likelihood is calculated as the conditional probability of encountering a property v_j given a property v_i in design change records, that is

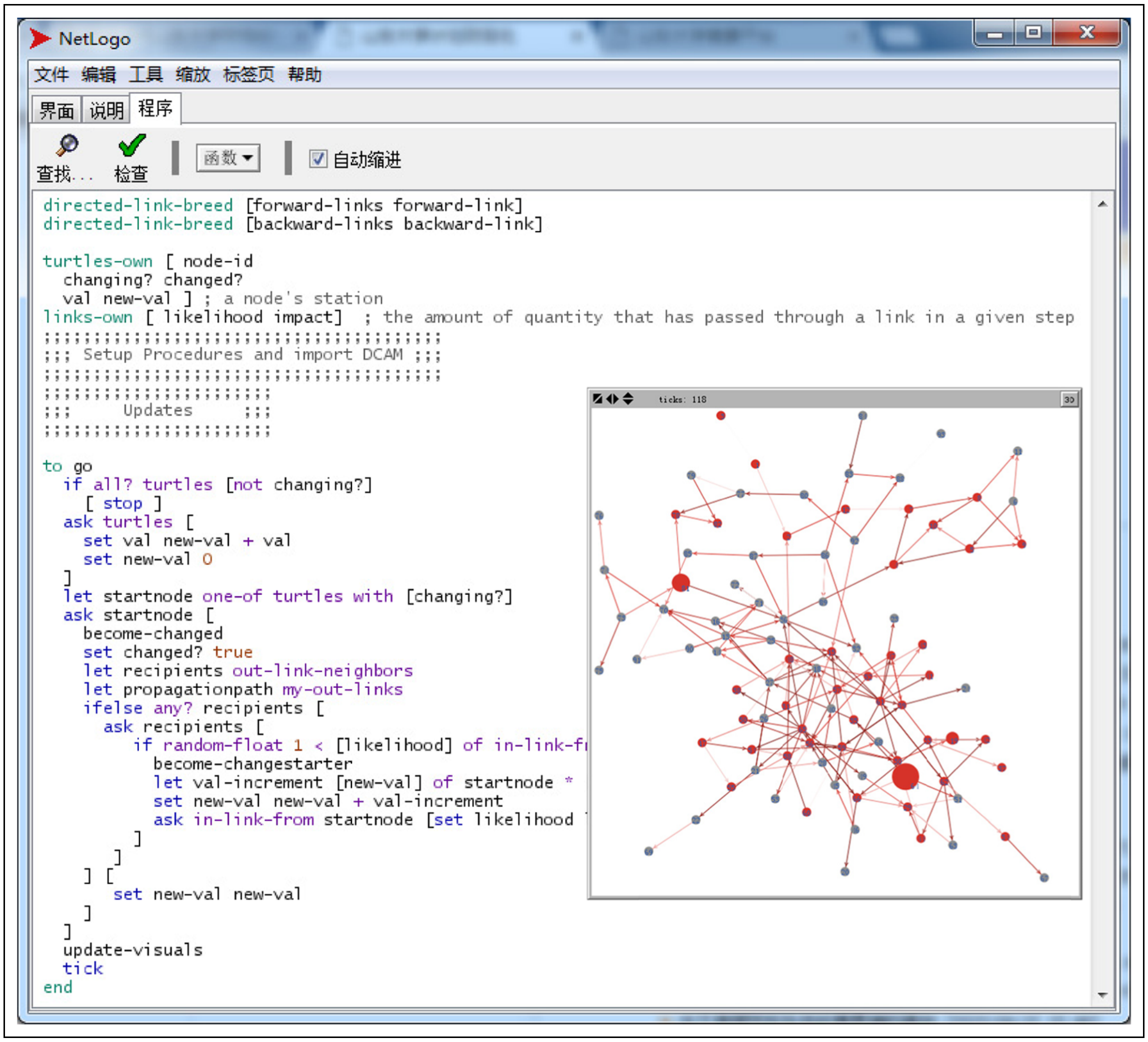
$$P_{i,j} = P\left(v_j|v_i\right) = \frac{P\left(v_i \cap v_j\right)}{P(v_i)} = P\left(v_i|v_j\right)\frac{P\left(v_j\right)}{P(v_i)} = P_{j,i}\frac{P\left(v_j\right)}{P(v_i)} \tag{8}$$

Normally, $P_{i,j}$ and $P_{j,i}$ are not equal since $P(v_i)$ is generally unequal to $P(v_j)$. If there is no linkage between the two nodes, $P_{i,j}$ equals 0. In the implementation of our method, the propagation likelihood is responsible for breaking the loop propagation with a certain probability.

As the multi-variations are propagating, the variables of agents and links update according to the change passing on it. The more the change records are used in training to evaluate $P_{i,j}$, the more accurate the simulation will be.

Implementation of change propagation using NetLogo

These kinds of simple agents have been produced by platforms such as MASON (http://cs.gmu.edu/~eclab/



```
directed-link-breed [forward-links forward-link]
directed-link-breed [backward-links backward-link]

turtles-own [ node-id
  changing? changed?
  val new-val ] ; a node's station
links-own [ likelihood impact]  ; the amount of quantity that has passed through a link in a given step
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;; Setup Procedures and import DCAM ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;     Updates      ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;

to go
  if all? turtles [not changing?]
    [ stop ]
  ask turtles [
    set val new-val + val
    set new-val 0
  ]
  let startnode one-of turtles with [changing?]
  ask startnode [
    become-changed
    set changed? true
    let recipients out-link-neighbors
    let propagationpath my-out-links
    ifelse any? recipients [
      ask recipients [
         if random-float 1 < [likelihood] of in-link-f
           become-changestarter
           let val-increment [new-val] of startnode *
           set new-val new-val + val-increment
           ask in-link-from startnode [set likelihood
         ]
      ]
    ] [
       set new-val new-val
    ]
  ]
  update-visuals
  tick
end
```

Figure 7. MAS platform—NetLogo.

projects/mason/), RePast (http://repast.sourceforge.net), SNMP (http://www.monfox.com/dsnmp/), and NetLogo (http://ccl.northwestern.edu/netlogo). In this article, NetLogo is chosen as the platform of our MAS. Specifically, NetLogo is a programmable modeling environment for simulating natural and social phenomena as shown in Figure 7. It is particularly well suited for modeling complex systems that develop over time. Developers can give instructions to hundreds or thousands of independent agents all operating concurrently. This makes it possible to explore the connection between the lower level behavior of agents and the collective behavior that emerges from the interactions of many agents.

The procedures of the multi-variation propagation simulation method can be summarized into four steps, that is, PREPARE, SETUP, UPDATE, and GO.

PREPARE step is responsible for creating the DCAM and identifying the multiple initial changed nodes. To define a DCAM appropriately, the links between nodes as well as the change impact and the propagation likelihood must be clearly specified. This step that prepares for importing the DCAM is implemented out of MAS. The temporal variation of a node is a variable involved in the agent. Then, the variables of an agent are defined as id, current value, new value, and change status (changed or changing). As mentioned above, the variables of a link are the propagation likelihood and the change impact.

In SETUP step, the DCAM is imported into MAS and each node is defined as an agent. This procedure reads a file that contains all the links. The file contains four columns separated by spaces. In this example, the links are directed. The first column contains the id of the node originating from the link, whereas the second column corresponds to the id of the node on the end of the link. Being the variables of each link, the last two columns are the propagation likelihood and the change impact, respectively.

Table 1. Pseudocode for multi-variation propagation simulation method based on the MAS.

Input: A DCAM O_D and the initial change nodes
Output: Predicted change propagation path

```
Algorithm://Change propagation prediction method based on MAS
Begin
  Evaluate the propagation likelihood $P_{ij}$ from design change records as the variable of each link of DCAM;
  Evaluate the change impact $I_{ij}$ for each link derived from the design function;
  Prepare the DCAM file in form of [$v_i$ $v_j$ $P_{ij}$ $I_{ij}$] in a row;
  Import the DCAM into NetLogo and define the agents;
  Set the initial change nodes;
  //Iteratively implement the change propagation process based on multi-agents
    Search the neighboring nodes along the out links of the change starters as the recipients until no changing node in the DCAM;
    If the possibility of change propagation is smaller than the propagation likelihood of the link
      Refresh the value of the connecting nodes according to the change impact;
      Update the propagation likelihood of this links;
      Set the current changing node as the change starter for the next propagation step;
    Endif
    Update the layout and the appearance of agents and links;
  End
  Return the predicted change propagation path;
End
```

MAS: multi-agent system; DCAM: design change analysis model.

After the DCAM is imported, the network should be layout. The UPDATE step is called to update the appearance of nodes and links. To help the users observe the variable of each agent, the size of nodes is scaled as its value is changed and the color of the link is brightened when the variation flows through it.

GO step, which is responsible for defining the rule of agents, is the main part of the simulation method. If the change conflict occurs in the propagation loop, it could be possible to jump out of the loop with the defined propagation likelihood. The value of each agent varies with the variation of the connecting agents according to the following equation

$$\begin{cases} c_j(t+1) = c_j(t) & P_{\text{rad}} > P_{i,j} \\ c_j(t+1) = c_j(t) + (c_i(t+1) - c_i(t))I_{i,j} & P_{\text{rad}} \le P_{i,j} \end{cases}, \quad i \in F_j \tag{9}$$

Searching the change propagation paths will not stop until it reaches the root nodes or leaf nodes of DCAM. Consequently, the illustration of the multi-variation propagation simulation method is concluded as shown in Table 1.

Case study and discussion

To apply the proposed method and verify its validity, take a gearbox as an example to study.

Case study

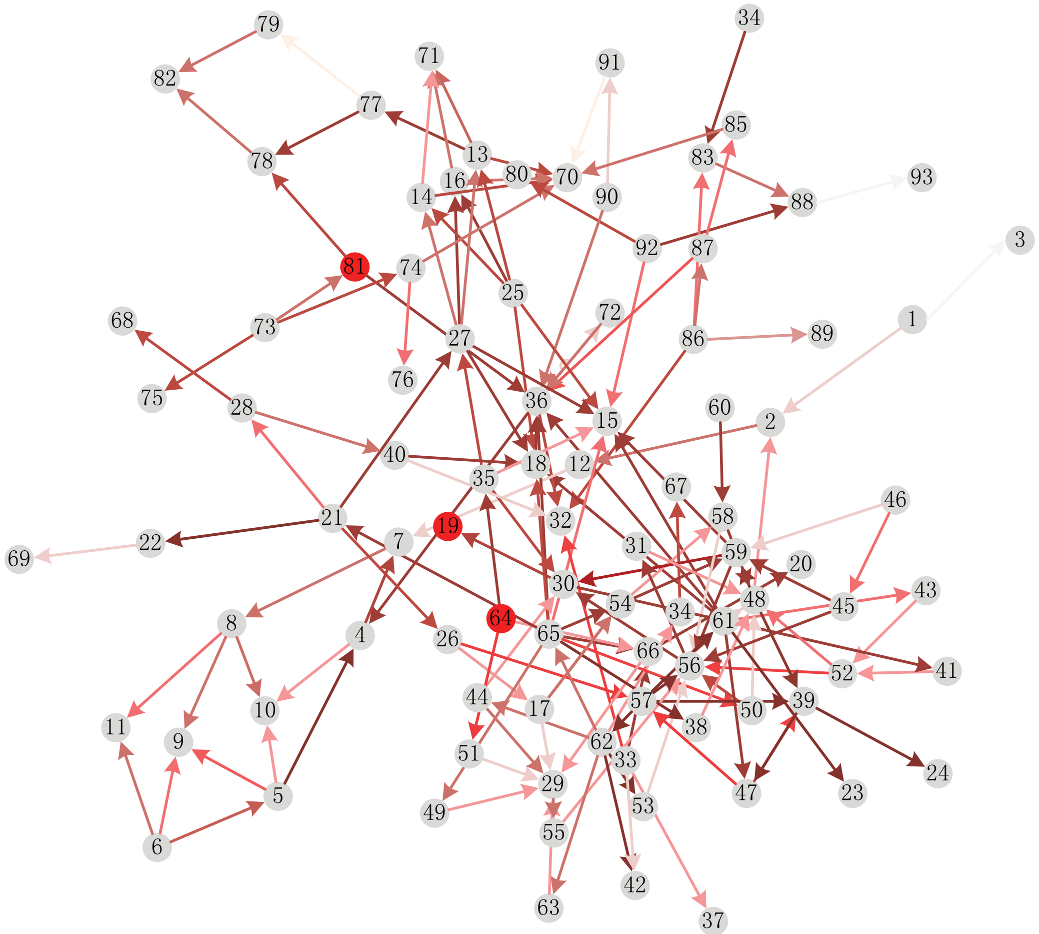
Gearbox offers a customized range of reduction ratios for a wide variety of applications in industries. Generally, a gearbox consists of gears, casing, shafts, seal, side covers, bearings, and keys. These components are characterized by being well fitted to make the parts closely interact with each other. A subtle change in the design of gearbox most probably results in the large-scale redesign. Therefore, designers need to evaluate the impact before implementing the design change.

The DCAM of gearbox is organized as a network and performs as the complex network theory. After following the design specification, the DCAM of gearbox is constructed as shown in Figure 8, which contains 93 nodes and 146 links. Please refer to Ma et al. (2016) for further details.

The propagation likelihood of each link of gearbox's DCAM is organized in a matrix as shown in Figure 9. The lightness of each element (i, j) is proportional to the magnitude of $P_{i,j}$ between nodes v_i and v_j (i.e. the darker elements represent the greater propagation probability). Most of the design properties are independent of others. The change impact of each link is organized in another matrix as shown in Figure 10. The propagation likelihood and the change impact as the inputs of the prediction method can be gained from the two matrices. By considering both the propagation likelihood matrix and the change impact matrix with the darker lightness, the critical nodes can be judged, which have the greater impact and higher possibility to diffuse the change. The variations prefer the nodes that are less critical in the DCAM to pass in terms of redesign based on the same performance and functions.

Simulation result and analysis

In this case study, the initial change is supposed to be triggered on the pitch diameter of gear, the bending



1 Low-speed train value i_{low}
2 Train value i_{total}
3 High-speed train value i_{high}
4 Power I P_{I}/ kW
5 Power II P_{II}/ kW
6 Power III P_{III}/ kW
7 Total power P_0/ kW
8 Overall efficiency η_{total}
9 Bearing efficiency η_{be}
10 Coupling efficiency η_{co}
11 Gear efficiency η_{ge}
12 Motor version V$_{mo}$
13 Coefficient A
14 Coefficient B
15 Face width b/ mm
16 Coefficient C
17 Base circle helix angle β_b/(°)
18 Pitch diameter d/ mm
19 Whole tooth depth h / mm
20 Heat treatment
21 Bending load factor K
22 Overload factor K_A
23 Contact load factor KF
24 Load-distribution factor $K_{F\alpha}$
25 Load-distribution factor $K_{F\beta}$
26 Load-distribution factor $K_{H\alpha}$
27 Load-distribution factor $K_{H\beta}$
28 Dynamic factor K_v
29 Gear material
30 Normal module m_n/ mm
31 Transverse module m_t/ mm
32 Rotation speed n/(r·min^{-1})
33 Number of load cycles N_L
34 Min. bending safety coefficient S_{Fmin}
35 Min. contact Safety coefficient S_{Hmin}
36 Tangential transmitted load T/(N·m)
37 Total working time t_h/ h
38 Train value u
39 Load-distribution factor $K'_{F\alpha}$
40 Speed v/(mm·s^{-1})
41 Geometry coefficient $Y_{F\alpha}$
42 Bending reliability coefficient Y_N
43 Stress correction coefficient $Y_{S\alpha}$
44 Size coefficient Y_X
45 Helix angle factor Y_{β}
46 Min. helix angle factor $Y_{\beta\min}$
47 Contact ratio coefficient Y_{ε}
48 Number of teeth z
49 Elastic coefficient Z_E
50 Pitch point coefficient Z_H
51 Contact reliability coefficient Z_N
52 Equivalent teeth number z_v
53 Helix angle factor Z_{β}
54 Contact ratio factor Z_{ε}
55 Transverse pressure angle α_t/(°)
56 Helix angle β /(°)
57 Equivalent transverse contact ratio $\varepsilon_{\alpha v}$
58 Transverse contact ratio ε_{α}
59 Overlap ratio ε_{β}
60 Total contact ratio ε_{γ}
61 Bending stress σ_f/ MPa
62 Allowable bending stress σ_F/ MPa
63 Bending fatigue limit σ_{Flim}/ MPa
64 Allowable contact stress σ_H/ MPa
65 Contact stress σ_h/ MPa
66 Contact fatigue limit σ_{Hlim}/ MPa
67 Single tooth meshing num. per cycle N$_{sc}$
68 Motor working condition
69 Meshing precision
70 Working condition
71 Check standard
72 Manufacturing accuracy
73 Allowable transmitted load T_{all}/ (N·m)
74 Allowable bearing stress σ_p/ MPa
75 Key version
76 Key material
77 Stress correcting factor α
78 Equivalent alternating stress σ_{-1b}/ MPa
79 Equivalent midrange stress σ_{0b}/ MPa
80 Shaft span b_{shaft}/ mm
81 Shaft diameter d_{shaft}/ mm
82 Shaft material σ_B/ MPa
83 Basic dynamic load C_r/ N
84 Determination factor e
85 Impulsive load coefficient f_d
86 Basic rating life L_{10h}/ h
87 Equivalent dynamic load P/ N
88 Bearing version
89 Bearing overload/ N
90 Coupling version
91 Allowable lifetime/ h
92 Shell width/ mm
93 Bore diameter/ mm

Figure 8. DCAM of a gearbox.

fatigue limit of gear, and the shaft span (corresponding to nodes v_{18}, v_{63}, and v_{80}). The volumes of initial variations are 0.2, 1.0, and 0.5, respectively. After simulating by MAS, three predicted change propagation paths are 18 → 19 → 30 → 35 → 27 → 15 → 92, 63 → 64 → 66 → 33 → 32, and 80 → 92, respectively, where

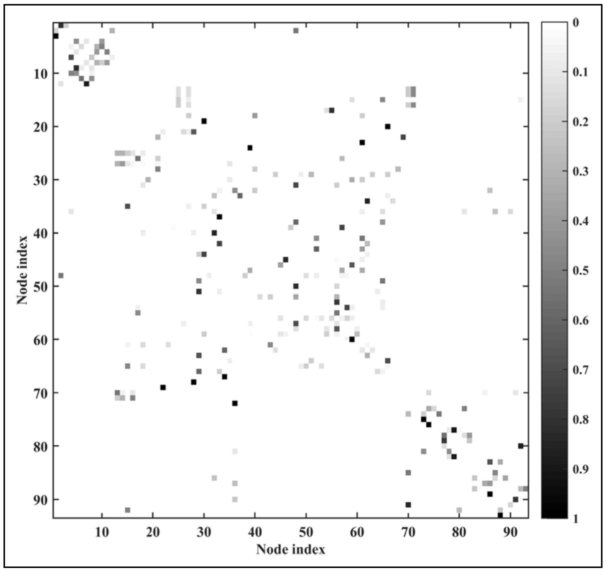


Figure 9. Matrix of propagation likelihood P_{ij} between two nodes.

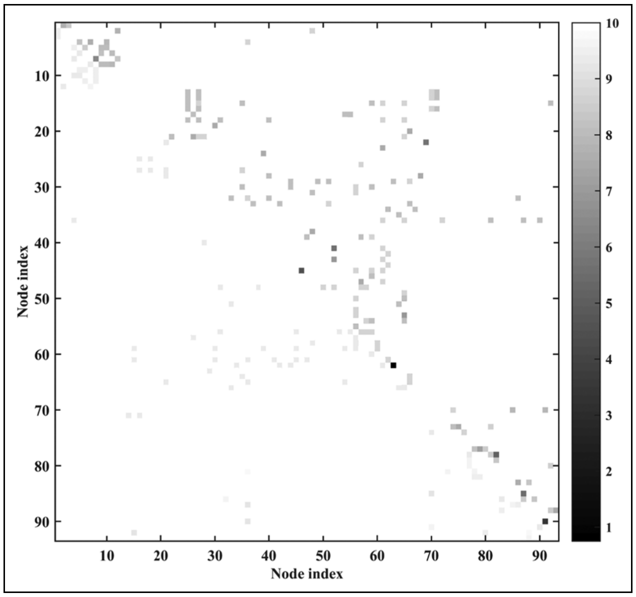


Figure 10. Matrix of change impact I_{ij} between two nodes.

the impacted nodes are scaled and highlighted in red as shown in Figure 11. During the simulation, the instant and the total number of the impacted nodes at each propagation step are shown in Figure 12. Since the initial change on node v_{18} is invoked, the propagation would successively impact the other design properties of the gear, that is, the tooth depth, the normal module, the minimum contact safety coefficient, the load distribution factor, and the face width of gear, and eventually changes the shell width of casting. Furthermore, the first and the third propagation paths converge at node v_{92}. Since both the changes enlarge the value of node v_{92} and no design constraint limits this node, no conflict occurs during the propagation of these three changes. The predicted change propagation paths are valid and can help the designers route the actual change flow.

To further verify the efficiency and effectiveness of the proposed method, we take change prediction method (CPM) as an example for comparison. The CPM as a classic method analyzes the change effort by combining the impact matrix and the likelihood matrix; the potential change propagation paths can be found by constructing a propagation tree based on the DSM and evaluated by gradually multiplying the propagation likelihood and impact along this tree. Since the DCAM of a gearbox contains several nodes and loops, it is too complex to construct the propagation tree for each node. In this article, the predicted change propagation paths of CPM are calculated by directly multiplying the propagation likelihood and the change impact along the shortest change propagation path between the initial change node and each leaf/root node. After the analysis using the modified CPM, the initial three changes would impact the other design properties with more possibility along these paths, that is, $18 \rightarrow 31 \rightarrow 56 \rightarrow 59 \rightarrow 39 \rightarrow 24$, $63 \rightarrow 29 \rightarrow 44 \rightarrow 62 \rightarrow 34 \rightarrow 67$, and $80 \rightarrow 92 \rightarrow 15 \rightarrow 59 \rightarrow 39 \rightarrow 24$, which are a little longer than the prediction from our method. Compared with our method, the predicted result from the CPM exposes two defects except for the computational complexity. One is that CPM cannot analyze the concurrent change propagations but the multi-variation propagation as liquid flows along several branches concurrently. Another defect is that the interactions between the multi-variations cannot be analyzed by the CPM, although the partially predicted propagation paths may be the same. Additionally, in the predicted change propagation paths resulting from CPM, the change on nodes v_{18} and v_{80} would eventually impact node v_{24}, that is, the load distribution factor, which is an empirical value with little changeability. In this case, the predicted propagation paths are less significant to guide the actual change propagation.

Conclusion

Since the multiple changes are frequent enough to be co-evoked during the concurrent design processes, the change propagation paths in different parallel processes may interact with each other because of the coupled properties in the complex product. The concurrent change propagations should be predicted before implementing; otherwise, the development time is delayed

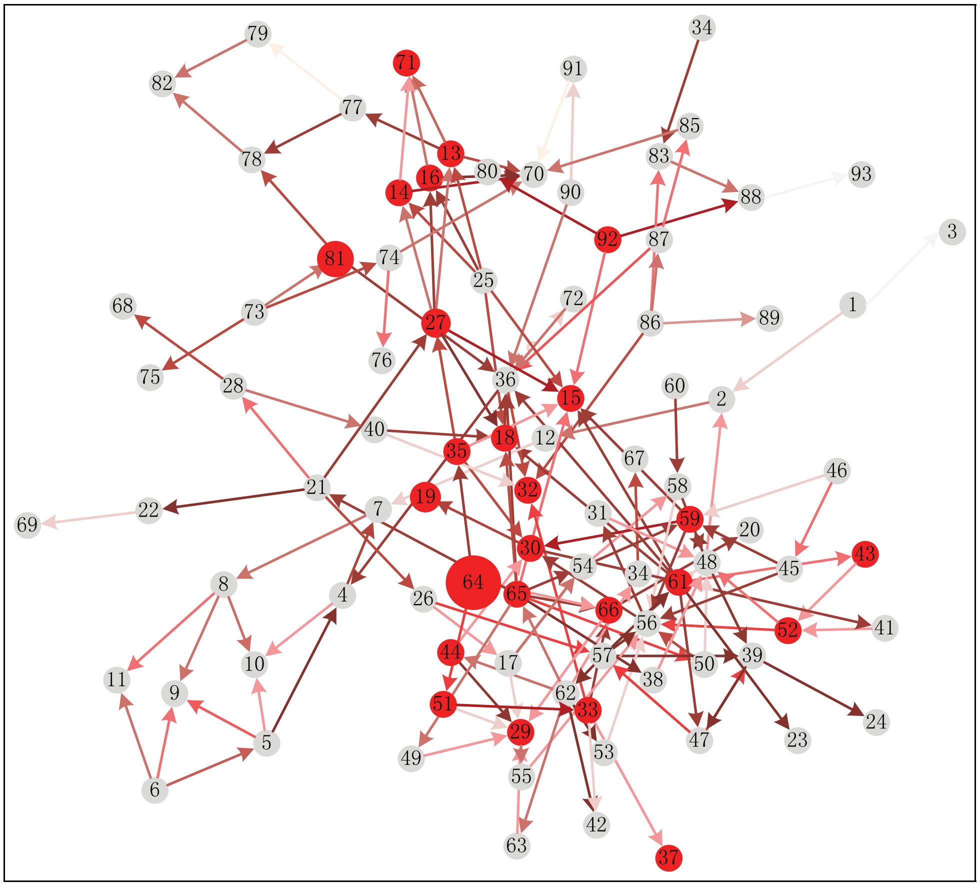


Figure 11. Change propagation path with the initial change triggered on node v_{81}.

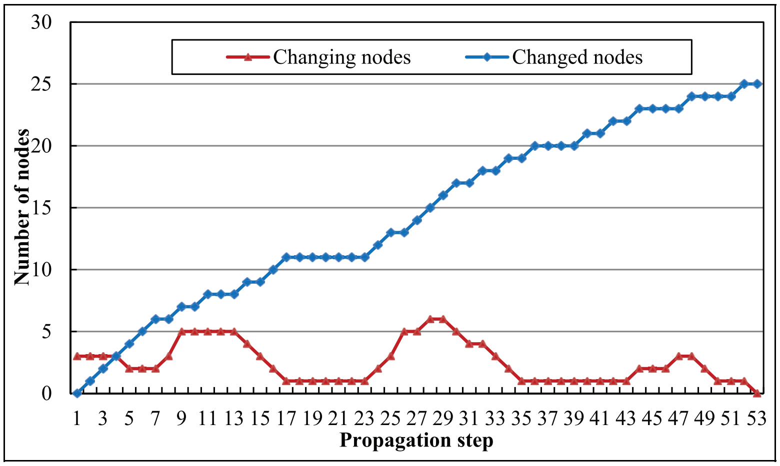


Figure 12. Statistics of the instant and the total number of the impacted nodes.

and the cost increases for the final outcome. In this article, a multi-variation propagation prediction approach is proposed to analyze the impact from the concurrent change propagations on mechanical product development in a convincing way. Compared with the previous research, the proposed method uses the DCAM as the propagation prediction foundation, which can reduce the man-made factors and improves the validity of the result; the change impact and the propagation likelihood as the impact metrics are defined objectively and quantitatively, which quantify the change propagation effects and improve the precision of change propagation prediction. Finally, the concurrent multi-variation propagations can be simulated by several programmed agents on the platform of MAS. Therefore, our method is capable of analyzing the concurrent change propagations since the multi-variations as liquid flowing propagate along several branches synchronously. In the future, we will focus our study on the multi-variation propagation mechanism for the complex mechanical design and the change propagation path optimization in concurrent design processes.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Natural Science Foundation of China (51175304), the China Postdoctoral Science Foundation (2015M572022), and the Distinguished Middle-Aged and Young Scientist Encourage and Reward Foundation of Shandong Province (BS2015ZZ004).

References

Ahmad N, Wynn DC and Clarkson PJ (2013) Change impact on a product and its redesign process: a tool for knowledge capture and reuse. *Research in Engineering Design* 24(3): 219–244.

Cheng H and Chu X (2012) A network-based assessment approach for change impacts on complex product. *Journal of Intelligent Manufacturing* 23(4): 1419–1431.

Chua DKH and Hossain MA (2012) Predicting change propagation and impact on design schedule due to external changes. *IEEE Transactions on Engineering Management* 59(3): 483–493.

Clarkson PJ, Simons C and Eckert C (2004) Predicting change propagation in complex design. *Journal of Mechanical Design* 126(5): 788–797.

Cohen T, Navathe SB and Fulton RE (2000) C-FAR, change favorable representation. *Computer-Aided Design* 32(5–6): 321–338.

Diprima M (1982) Engineering change control and implementation considerations. *Production and Inventory Management* 23(1): 81–87.

Duan G and Wang Y (2013) QCs-linkage model based quality characteristic variation propagation analysis and control in product development. *International Journal of Production Research* 51(22): 6573–6593.

Eckert C, Clarkson PJ and Zanker W (2004) Change and customisation in complex engineering domains. *Research in Engineering Design* 15(1): 1–21.

Fei G, Gao J, Owodunni O, et al. (2011) A method for engineering design change analysis using system modelling and knowledge management techniques. *International Journal of Computer Integrated Manufacturing* 24(6): 535–551.

Gero JS (1990) Design prototypes: a knowledge representation schema for design. *AI Magazine* 11(4): 26.

Hamraz B, Caldwell NHM and Clarkson PJ (2013a) A holistic categorization framework for literature on engineering change management. *Systems Engineering* 16(4): 473–505.

Hamraz B, Caldwell NHM and Clarkson PJ (2013b) A matrix-calculation-based algorithm for numerical change propagation analysis. *IEEE Transactions on Engineering Management* 60(1): 186–198.

Koh ECY, Caldwell NHM and Clarkson PJ (2012) A method to assess the effects of engineering change propagation. *Research in Engineering Design* 23(4): 329–351.

Lee H, Seol H, Sung N, et al. (2010) An analytic network process approach to measuring design change impacts in modular products. *Journal of Engineering Design* 21(1): 75–91.

Li N, Li X, Shen Y, et al. (2015) Risk assessment model based on multi-agent systems for complex product design. *Information Systems Frontiers* 17(2): 363–385.

Li S and Chen L (2014) Identification of clusters and interfaces for supporting the implementation of change requests. *IEEE Transactions on Engineering Management* 61(2): 323–335.

Li W and Moon YB (2012) Modeling and managing engineering changes in a complex product development process. *The International Journal of Advanced Manufacturing Technology* 63(9–12): 863–874.

Li Y and Zhao W (2014) An integrated change propagation scheduling approach for product design. *Concurrent Engineering* 22(4): 347–360.

Li Y, Zhao W and Ma Y (2016) A shortest path method for sequential change propagations in complex engineering design processes. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 30(1): 107–121.

Li Y, Zhao W and Shao X (2012) A process simulation based method for scheduling product design change propagation. *Advanced Engineering Informatics* 26(3): 529–538.

Ma S, Jiang Z and Liu W (2016) A design change analysis model as a change impact analysis basis for semantic design change management. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*. Epub ahead of print 17 February. DOI: 10.1177/0954406216633035.

McIntosh KG (1995) *Engineering Data Management: A Guide to Successful Implementation*. New York: McGraw-Hill.

Mehta C, Patil L and Dutta D (2013) An approach to determine important attributes for engineering change evaluation. *Journal of Mechanical Design* 135: 0410034.

Morkos B, Shankar P and Summers JD (2012) Predicting requirement change propagation, using higher order design structure matrices: an industry case study. *Journal of Engineering Design* 23(12): 905–926.

Ouertani MZ (2008) Supporting conflict management in collaborative design: an approach to assess engineering change impacts. *Computers in Industry* 59(9): 882–893.

Pasqual MC and De Weck OL (2012) Multilayer network model for analysis and management of change propagation. *Research in Engineering Design* 23(4): 305–328.

Prasad B (2002) Building blocks for a decision-based integrated product development and system realization process. *Systems Engineering* 5(2): 123–144.

Reddi KR and Moon YB (2011) System dynamics modeling of engineering change management in a collaborative environment. *International Journal of Advanced Manufacturing Technology* 55(9–12): 1225–1239.

Shankar P, Morkos B and Summers JD (2012) Reasons for change propagation: a case study in an automotive OEM. *Research in Engineering Design* 23(4): 291–303.

Tang DB, Yin LL, Wang Q, et al. (2016) Workload-based change propagation analysis in engineering design. *Concurrent Engineering* 24(1): 17–34.

Wynn DC, Caldwell NHM and John Clarkson P (2014) Predicting change propagation in complex design workflows. *Journal of Mechanical Design* 136(8): 81009.

Yang F and Duan G (2012) Developing a parameter linkage-based method for searching change propagation paths. *Research in Engineering Design* 23(4): 353–372.

Author biographies

Songhua Ma, born in 1985, is currently a research assistant at Key Laboratory of High Efficiency and Clean Mechanical Manufacture Shandong University of Ministry of Education, Shandong University, China. She received her PhD from Tsinghua University, China, in 2014 and her bachelor degree got from Shandong University, China, in 2008. Her research interests include CAD/CAM and knowledge management.

Zhaoliang Jiang, born in 1971, is currently a professor and PhD supervisor at Key Laboratory of High Efficiency and Clean Mechanical Manufacture, Ministry of Education, Shandong University, China.

Wenping Liu, born in 1973, is currently an associate professor at Key Laboratory of High Efficiency and Clean Mechanical Manufacture, Ministry of Education, Shandong University, China.