Research in Engineering Design (2021) 32:507–522
https://doi.org/10.1007/s00163-021-00369-6

ORIGINAL PAPER

Modularization for the complex product considering the design change requirements

Yupeng Li[1] · Yongbo Ni[1] · Na Zhang[1] · Zhenhua Liu[2]

Received: 15 October 2020 / Revised: 4 May 2021 / Accepted: 2 June 2021 / Published online: 15 June 2021

Abstract

Driven by the evolving customer requirements and the advancement of key technologies, design changes broadly exist in the lifecycle of the complex product. And as a popular engineering management strategy, the modularization strategy has been widely applied in the research and development process of complex products. However, most of the existing modularization methods do not consider the issue of design change management. Under this circumstance, the "avalanche effect" of the design change propagation might be magnified due to the inappropriate modular structure. Thus, to decrease the effect of the design change propagation, a novel modularization method of the complex product is proposed incorporating the modularity and the scope of design change propagation (SDCP). Firstly, considering the functional and physical relationship between components, the correlation matrix that is the adjacency matrix of the related weighted and directed network model is constructed. Secondly, the indexes of modularity and SDCP are defined based on the predetermined network model and correlation matrix, respectively. Thirdly, taking the modularity and the SDCP as the optimization objectives, a bi-objective optimization model is built for the modularization of the complex product, and then the model is solved by the Non-dominated Sorting Genetic Algorithm-II (NSGA-II). Finally, the modularization of the cab for a specific electronic sanitation vehicle is implemented as the case study to expound the utility and effectiveness of the proposed methodology.

Keywords Product modularization · Modularity · SDCP · Bi-objective optimization · NSGA-II

1 Introduction

As a country's high-end equipment, complex products play an increasingly important role in the development of modern society, reflecting a country's comprehensive national strength and scientific competitiveness (Prencipe 2000). Complex products such as large-scale construction machinery, high-speed trains, and carrier rockets always have high technological content, long lead-time of research and development (R&D), and complicated coupling relations among components. Due to the continuously changing internal factors (advancement of key technologies) and external requirements (evolving customer requirements), the relative functions and physical structures of a complex product are also changing in the product lifecycle (Morkos et al. 2012). This phenomenon is termed the "design change" in the field of product R&D. Moreover, with the design change requirements exceed the absorption capacity of design margins (Eckert et al 2020; Brahma and Wynn 2020), it will successively propagate from the original component to other ones and then the "avalanche effect" will occur on account of the complex coupling relations among components (Eckert et al. 2004; Braha and Yassine 2003; Braha and Bar-Yam 2004a, b). The propagation will lead to the high cost and long response lead-time for design change, which will result in a negative impact on the cost control and customer satisfaction warranty. Thus, the enterprise should pay considerable attention to the management of design change by controlling the scope of design change propagation (SDCP) to reduce the unnecessary extra cost.

Benefiting from the advantages of concurrent engineering, the modularization strategy has been widely applied by enterprises to enhance the R&D efficiency of complex products (Braha and Yassine 2003; Li et al. 2017a, b). Product

✉ Yupeng Li
ypeng_li@163.com

[1] Department of Industrial Engineering, School of Mines, China University of Mining and Technology, Xuzhou 221116, Jiangsu, China

[2] School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

modularization is driven by many factors, like structural and functional relationships, material efficiency, supplier constraints, and maintenance requirements (Yu et al. 2011; Ji et al. 2013; Salvador and Villena 2013; Sinha et al. 2020), wherein structural and functional relationships are the common and fundamental factors for modular design. And module identification is the foundation for modularization since its result will directly impact the effectiveness and cost of product R&D. Traditionally, most of the existing studies on module identification only focus on the "modularity" of the identified modular product structure (Gershenson et al. 2003; Guo and Gershenson 2007). However, few studies consider the design change management in the modularization process. When a component changes in a module, it will easily propagate to other modules and cause multiple system-wide cascades due to the interfaces among modules (Braha and Bar-Yam 2007). Thus, intuitively, there is a claim that "the design change should be limited to as few modules as possible" (Li and Chen 2014). In other words, the design change requirements should be considered in the modularization process for a complex product to limit the predictable SDCP.

Motivated by the above discussion, a novel modularization method incorporating modularity and SDCP is proposed. First, the representation models of the complex product include the correlation matrix and the weighted and directed network model are constructed. Second, the indexes of modularity and SDCP are defined based on the predetermined network model and correlation matrix, respectively. Third, a bi-objective model that takes modularity and SDCP as the optimization objectives is constructed. Finally, a Non-dominated Sorting Genetic Algorithm-II (NSGA-II) is designed to solve the model.

The rest of this paper is organized as follows. The related works are reviewed in Sect. 2. The proposed approach is introduced in Sect. 3. A case study is presented in Sect. 4. The conclusions and future works are summarized in Sect. 5.

2 Literature review

The two important issues that are product modularization and design change propagation are incorporated in this study, and the literature on them will be reviewed in this section.

2.1 Product modularization

Module identification is the foundation to implement the modularization strategy. Research attention has been paid to many issues about modularization such as product modeling methods and optimization objectives.

From the viewpoint of the product modeling method, the research on modularization can be divided into two categories that are matrix-based method and network-based method. The design structure matrix (DSM) and complex network are the typical models, respectively (Braha and Yassine 2003; Braha and Bar-Yam 2004a, b). For instance, Beek et al. (2010) used DSM to represent the functional relations among components and then employed the K-means algorithm to realize the module identification. Additionally, Li et al. (2017a) developed a hybrid product modular analysis approach to cluster components with DSM. Likewise, recent studies on modularization have explored the advantages of complex networks. The weighted network, weighted and directed network were successively constructed to establish the modularization for complex products (Li et al. 2014, 2017b). Most recently, Zhang et al. (2019) employed the weighted complex network to model the wind turbine and then used a community detection algorithm to realize the clustering for components. Inspired by these researches, the principle of community detection in the complex network theory can effectively deal with the modularization problem for the complex product.

Modularization is essentially an optimization problem in which the optimization objective (or the criterion of module identification) is a critical issue. For example, Braha and Maimon (1998) proposed a modularization criterion that is the higher coherence in the same module and the lower coupling between modules, which is called "modularity" (Gershenson et al. 2003). Then, taking modularity as the criterion, Cheng et al. (2018) developed the density-based spatial clustering of applications with noise to establish modularization. Afterward, the overall modularity index was defined to optimize the granularity and numbers of modules (Algeddawy and Elmaraghy 2013). Besides, the overall top-level distance (Li et al. 2008), and the minimum description length (Yu et al. 2007) are proposed as modularity assessment indices, respectively. As stressed in the last paragraph, the complex network has been investigated as a powerful means for modularization. The modularity of a complex network (Newman 2004) was employed by scholars in the recent decade. The modularity of the community detection in the complex network theory was originally introduced by Li et al. (2014) to the modularization of complex products. Thereafter, Zhang et al. (2019) employed the modified GN community detection algorithm to identify the modules of complex mechanical products. Li et al. (2019b) developed a hierarchical clustering algorithm to detect modules, which indicates the superiority of network modularity in modularization. Besides, directed network modularity (Kim et al. 2010) was applied for functional module identification at the conceptual design stage of the complex product (Li et al. 2017b).

According to the aforementioned studies, there has been growing academic attention in product modularization, but still, some issues need to be improved. That is, previous studies have not sufficiently considered some other optimization objectives or criteria, especially for the design change requirements-related criterion. The design change can not be avoided in the R&D process of a complex product; thus, it is significant to establish a modular architecture that can effectively support the management of design change for complex products.

2.2 Design change propagation

Design changes are inevitable in the R&D process of complex products. Jarratt et al. (2011) has combed through the reasons for design changes, including but not limited to satisfy the emerging customer requirements, improve the quality of products, change the suppliers, et al. Due to the direct or indirect dependency between components and systems, changes of one component or system can propagate to other ones, and this process is called design change propagation. Considerable attention has been paid to the design change propagation. Yassine et al. (2003) developed the work transformation method based on DSM to formulate the dynamics of the design project. Li and Chen (2014) employed the domain mapping matrix to analyze the change propagation from functions to components. Braha and Bar-Yam (2004a, b) applied the complex networks to analyze design change propagation and they found the engineering network is dominated by few highly "central" nodes which cause that the network is highly vulnerable when perturbations occur at them. They also demonstrated that reducing the coupling between a highly connected node and other nodes will result in attenuating the effect of design change propagation.

Scholars have conducted many studies on change propagation prediction to manage and control the design change propagation, especially for the prediction of change propagation path and change propagation influence. For instance, Cohen et al. (2000) proposed a change favorable representation to analyze the change propagation paths from the source entity to the target entity. Consider the multi-change requirements, Ullah et al. (2018) developed an optimization model of change propagation risk to predict the change propagation paths. Li et al. (2019a) proposed the change propagation intensity to forecast the change propagation paths. Likewise, the research on change influence prediction is significant. To predict the influence of design change, Clarkson et al. (2014) analyzed the direct and indirect dependencies among components via the change propagation tree. Martin and Ishii (2002) used DSM to analyze the effect of a design change on components for product variety design. Apart from the above studies, Li and Chen (2014) defined the SDCP to obtain the dynamic clustering of components. This research is an important attempt to combine modularization with design change management, but it assumes that the modular structure of the product will change dynamically with the source of the design change, which is not conducive to product development management. Besides, although the interfaces are identified, the effect of design changes on the interfaces is not considered in SDCP, and the relationships among components are only modeled as the Boolean matrix. In general, there are massive studies about design change management, but some drawbacks still exist. The current research cannot accurately quantify the SDCP, and how to effectively determine and control the SDCP in modular products deserves further study.

In summary, few studies focused on the interaction between modularization and design change management. Taking this in mind, the contribution of this study is twofold as follows.

(1) The bi-objective optimization model for modularization of the complex products incorporating modularity and SDCP is proposed. The obtained product modular structure can balance the product modularity and the flexibility for controlling the design change propagation.
(2) A modified SDCP index is proposed considering the punishment of interfaces between modules. Except for measuring the scope of change propagation, the modified SDCP index can also be used to obtain the key interfaces which are employed as the feedback to effectively control the SDCP.

3 Construction and solution of the bi-objective modularization model

The main idea of the modularization method for the complex product is shown in Fig. 1. Above all, considering the functional and structural relationships among components, the correlation matrix (adjacency matrix of the weighted and directed network model) is constructed. Second, modularity and SDCP are defined. Next, a bi-objective model is established considering modularity and SDCP. Finally, the NSGA-II is developed to solve this model.

3.1 Determine the comprehensive correlation matrix of components

The relations among components are the input of product modularization, and it is established from several aspects. In this study, the functional and structural relations (Yu et al. 2011) among components are considered. The quantification rules of the functional and structural relation are listed in Tables 1 and 2, respectively. The structural relations among

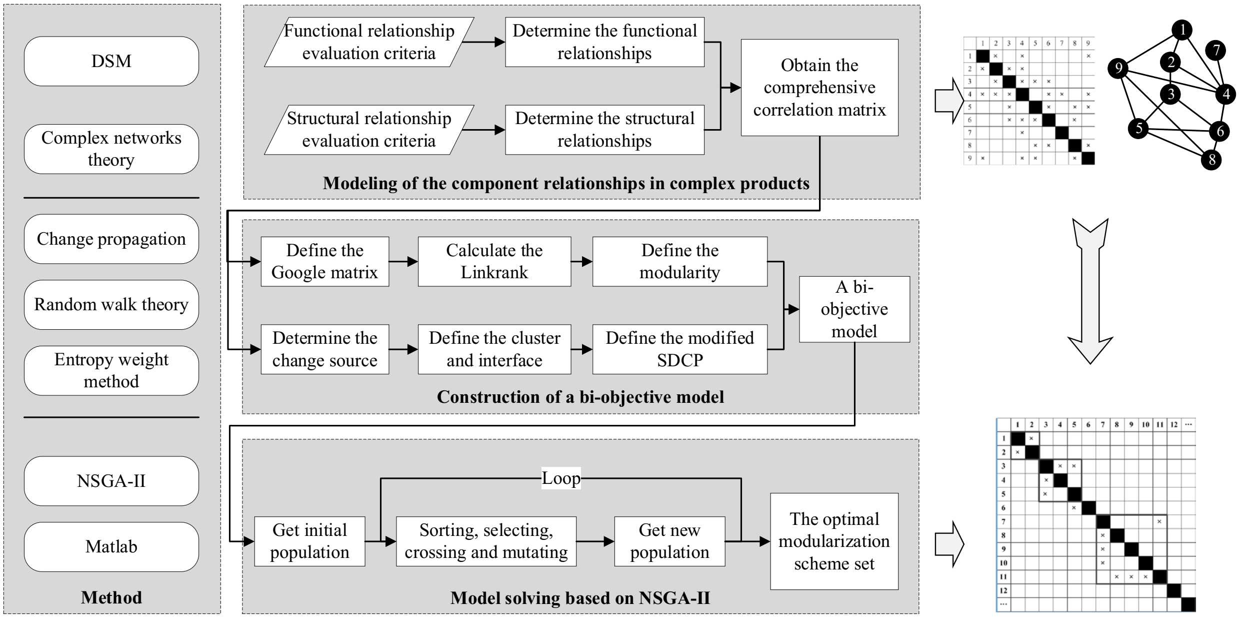
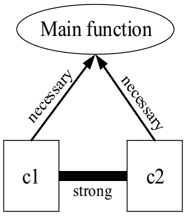
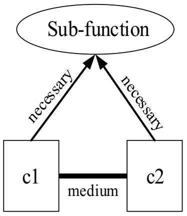
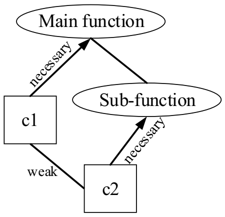
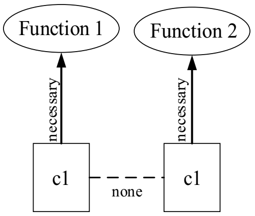

DSM
Complex networks theory
Change propagation
Random walk theory
Entropy weight method
NSGA-II
Matlab
Method
Functional relationship evaluation criteria
Determine the functional relationships
Structural relationship evaluation criteria
Determine the structural relationships
Obtain the comprehensive correlation matrix
Modeling of the component relationships in complex products
Define the Google matrix
Calculate the Linkrank
Define the modularity
Determine the change source
Define the cluster and interface
Define the modified SDCP
A bi-objective model
Construction of a bi-objective model
Loop
Get initial population
Sorting, selecting, crossing and mutating
Get new population
The optimal modularization scheme set
Model solving based on NSGA-II
1
2
3
4
5
6
7
8
9
10
11
12
...

Fig. 1 Technical framework of the proposed method

Table 1 Scaling of the functional relationship

Strength	Strong	Medium	Weak	None
Score	1	0.6	0.2	0
Types	 Main function necessary necessary c1 c2 strong	 Sub-function necessary necessary c1 c2 medium	 Main function necessary Sub-function c1 weak c2 necessary	 Function 1 Function 2 necessary necessary c1 none c1
Description	Two components are both necessary to realize the main function.	One component is necessary to realize the main function, the other is necessary to realize the subordinate function of this main function.	Two components are necessary to realize the same subordinate function.	Two components are necessary to realize different functions.

components are symmetry while the functional relations are asymmetry due to the directions of functional flow among components. The comprehensive correlation matrix (CCM) of components can be defined as the adjacency matrix of a complex network (Braha and Yassine 2003). The CCM is represented as

where **A** is the CCM (or the adjacency matrix of a network model), $0 \leq A(i, j) \leq 1$ is the comprehensive relation between component (node) i and j. Remarkably, the node in the adjacency matrix of a network mentioned in the following represents the component. Specifically, $A(i, j)$ can be expressed as

$$\mathbf{A} = \begin{cases} A(i,\ j), & \text{if there is a relation between component (node) } i \text{ and } j, \\ 0, & \text{others,} \end{cases} \tag{1}$$

Table 2 Scaling of the structural relationship

Strength	Score	Contact pattern	Combination pattern	Degree of freedom
Very strong	1	Multi-planes contact	Non-disassemble	1
Strong	0.8	Multi-points contact	Dismountable	2
Medium	0.6	Single plane	Screw in	3
Weak	0.4	Line contact	Insert	4
Very weak	0.2	Point contact	Flat	5

$$A(i,j) = w_{\mathrm{f}} \times A(i,j)^{\mathrm{f}} + w_{\mathrm{s}} \times A(i,j)^{\mathrm{s}}, \tag{2}$$

where $A(i, j)^{\mathrm{f}}$ is the functional relationship, $A(i, j)^{\mathrm{s}}$ is the structural relationship. w_{f} and w_{s} represent the weight of the functional and structural relationship, respectively ($w_{\mathrm{f}} > 0$, $w_{\mathrm{s}} > 0$, and $w_{\mathrm{f}} + w_{\mathrm{s}} = 1$). $A(i, i) = 1$, $(i, j = 1, 2, \ldots, n)$, n is the number of components.

3.2 Objectives of modularization

Compared with the traditional modularization methods which take modularity as the optimization objective, a bi-objective modularization model which simultaneously takes the modularity and SDCP as the optimization objectives is proposed. The “modularity” based on LinkRank (Kim et al 2010) is an effective criterion to evaluate the quality of modularization. And compared with other modularization indexes, it takes the information on directions of relationship between components into consideration which is suitable for module partition in both undirected and directed product networks. As for the index of “SDCP”, it uses clustered structures to group components in one or few clusters and identify interfaces to control the propagation scope of design change (Li and Chen 2014), which provides a relatively simple approach to analyze the change propagation at the cluster level than that at the component level.

3.2.1 Modularity

Modularity based on LinkRank (Kim et al 2010) is adopted for the relationship between components is direct. The steps of calculating the modularity based on LinkRank are four steps.

Step 1. Obtain the Google matrix through the network adjacency matrix. Google matrix is the probability matrix for the random walk process and it is the basis to analyze the importance of nodes in the network.

$$G_{ij} = \alpha \frac{w_{ij}}{w_i^{\mathrm{out}}} + \frac{1}{n}(\alpha a_i + 1 - \alpha), \tag{3}$$

where G_{ij} is the probability that a random walker on node i moves toward node j in the next random walk step. α is the damping coefficient representing the probability of following the hyperlinks when users arrive at one specific webpage. w_{ij} is an element in the network adjacency matrix, w_i^{out} is the out-degree of node i. And a_i is a variable for modeling whether node i is a dangling node. If node i is a dangling node, $a_i = 1$, else, $a_i = 0$.

Step 2. Calculate the PageRank based on Google matrix.

PageRank, also known as the importance of a node, indicates the probability of randomly opening a hyperlink on a webpage to reach a specific page (Brin and Page 2012), and it is calculated by

$$\boldsymbol{\pi}^{\mathrm{T}} = \boldsymbol{\pi}^{\mathrm{T}} G, \tag{4}$$

where $\boldsymbol{\pi}^{\mathrm{T}}$ is the stationary row vector of matrix $\mathbf{G}$, which is also called the PageRank vector, and matrix $\mathbf{G}$ is the Google matrix.

Step 3. Calculate the LinkRank between nodes.

$$L_{ij} = \pi_i G_{ij}, \tag{5}$$

where L_{ij} is used to measure the importance of the edge between node i and j,π_i is the probability of visiting node i by random steps in the stationary state.

Step 4. Calculate the modularity considering LinkRank.

$$Q_{\mathrm{LinkRank}} = \sum_{i,j} [L_{ij} - \pi_i \pi_j] \delta(c_i,\ c_j), \tag{6}$$

where L_{ij} is the LinkRank between node i and node j, and $\pi_i \pi_j$ represents the probability that node i randomly walks to node j. If node i and node j are in the same community (module), $\delta(c_i,\ c_j) = 1$, otherwise, $\delta(c_i,\ c_j) = 0$.

Taking the module partition scheme in Fig. 2 as an example, the PageRank vector of the 9 nodes is $\boldsymbol{\pi}$ = [0.1038, 0.1038, 0.1395, 0.0920, 0.1157, 0.1098, 0.1217, 0.1098, 0.1038], according to Eqs. (3) and (4). Then based on the PageRank, the LinkRank of each edge can be calculated according to Eq. (5) as

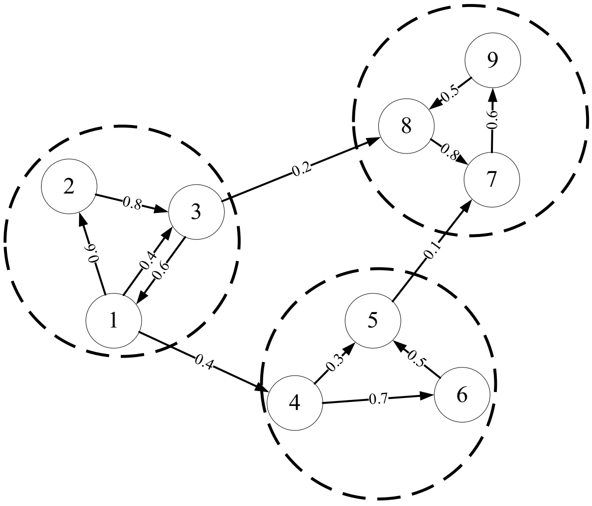


Fig. 2 Three modules in a complex network

$$
L = \begin{bmatrix}
0 & 0.6023 & 0.0415 & 0.0415 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.0831 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.0837 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0279 & 0 \\
0 & 0 & 0 & 0 & 0.0276 & 0.0644 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.0116 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.0549 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0730 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.0878 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0519 & 0
\end{bmatrix}
$$

Finally, the modularity of this scheme with three modules based on LinkRank is calculated, and it is 0.41.

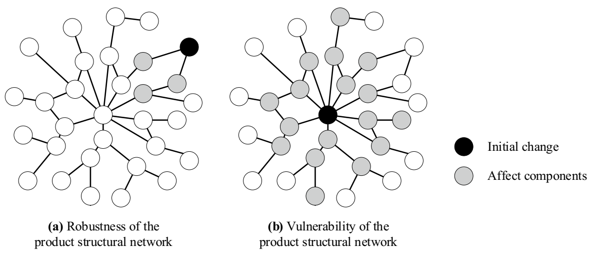
3.2.2 Scope of the design change propagation (SDCP)

Design change propagation will cause multiple system-wide cascades (Braha and Bar-Yam 2007; Braha 2016) and increase the cost of redesign. To manage and control the SDCP, the strategies of modularization and modified interfaces are developed (Braha and Yassine 2003).

(1) Selection of target element(s)

Although any component may change in the design process, not all the changes will have a serious impact on the product structure. The product structure is robust under random attacks (errors, failures, or design changes) and vulnerable under target attacks (Braha and Bar-Yam 2007; Park and Kremer 2019). As shown in Fig. 3, the change of the random nodes has little effect on the entire network structure. However, the change of "hub" (core) nodes may affect most of the network structure. Besides, only the components which are sensitive to the dynamic internal and external factors may tend to change and then propagate to others. To control the design change through a specific modularization scheme, it is necessary to select the target elements which are both influential and easy to change.

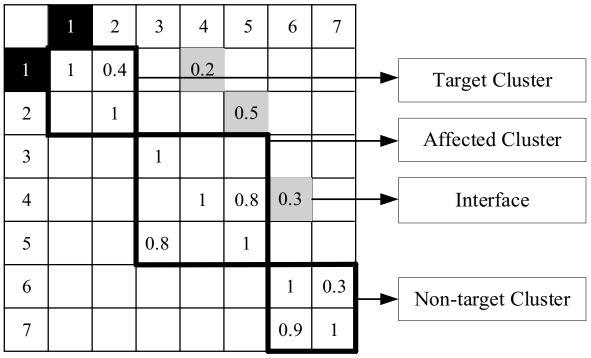
The target elements (TEs) are regarded as the change sources and the affected elements (AEs) are named for the components which have a direct correlation with the TEs. Meanwhile, the components that have no direct correlation with the TEs are defined as the non-target elements (NEs). As shown in Fig. 4, let component 1 as a TE. Then component 2 and component 4 are the



(a) Robustness of the product structural network

(b) Vulnerability of the product structural network

Fig. 3 Illustration of the robustness and vulnerability of the product structural network



	1	2	3	4	5	6	7
1	1	0.4		0.2			
2		1			0.5		
3			1				
4				1	0.8	0.3	
5			0.8		1		
6						1	0.3
7						0.9	1

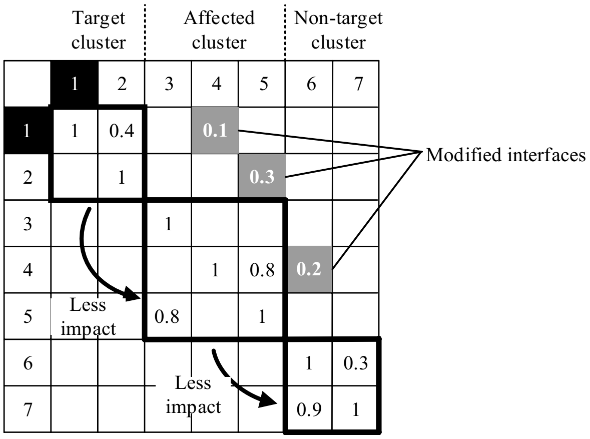
Fig. 4 Clusters and interfaces

AEs, and component 3, component 5, and component 6 are the NEs.

(2) Definition of the cluster and the interface

Different combinations of components will form different clusters, and they can be defined as a target cluster, an affected cluster, or a non-target cluster that are connected through interfaces (Li and Chen 2014).Definition 1 Target cluster. The cluster includes at least one TE. For example, in Fig. 4 the cluster containing component 1 and component 2 is a target cluster since component 1 is a target element.Definition 2 Affected cluster. The cluster contains an AE but does not have a TE. In Fig. 4, the cluster that includes an AE (component 4) and does not have a TE is the affected cluster.Definition 3 Non-target cluster. The cluster only has the NE(s), such as the cluster that is formed of component 6 and component 7 (two NEs).Definition 4 Interface. It's the relationship between clusters. As shown in Fig. 4, the interfaces are marked in grey.

3. Calculation of the SDCP



	1	2	3	4	5	6	7
1	1	0.4		0.1			
2		1			0.3		
3			1				
4				1	0.8	0.2	
5			0.8		1		
6						1	0.3
7						0.9	1

Fig. 5 Modify interfaces between clusters

As shown in Fig. 4, when the target element changes, it will propagate to other components within the same module and even propagate to other modules through interfaces, which results in the complexity of design change management. In this study, elements in the same cluster are treated collectively and concurrently as one unit in the implementation of change requests (Li and Chen 2014). The elements of the same cluster should be closely related, such as the affected cluster shown in Fig. 4, the relationship of component 5 to component 3 is strong, as well as the influence from component 4 to component 5. Yet, there is no correlation between component 3 and component 4. For clustering components in a group with a close relationship and reducing the situation with non-relationship, the scope of change propagation in a cluster is defined as:

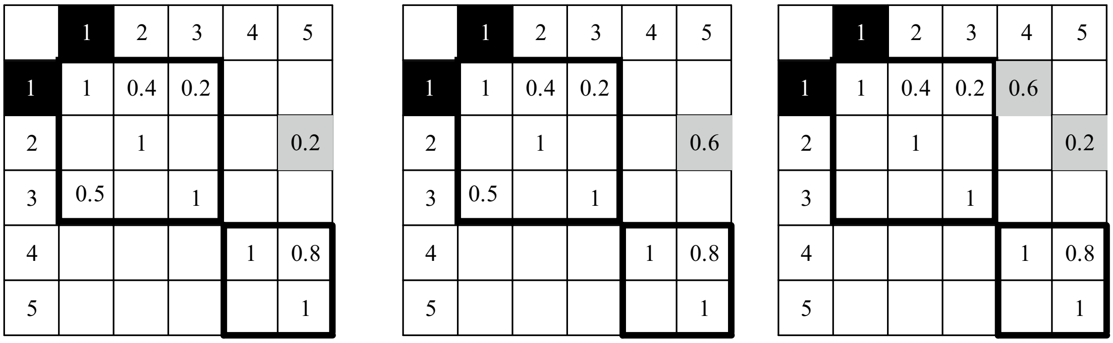
$$S = \sum_i \sum_j A(i, j) + w_0 \times m, \tag{7}$$

where S is the changing scope within a cluster, i, j are the components in the cluster, $A(i, j)$ is the relationship between i and j, m represents the number of blank grids in a cluster, w_0 is a minimum weight of m, which is set to be 0.05.

To mitigate the cascading effect of change propagation, the interfaces can be modified by reducing the relationship of interfaces between components or modules (Braha and Bar-Yam 2007). As shown in Fig. 5, the relationships of interfaces between modules are reduced. So, when the design change propagates from the target cluster to the affected cluster, it will have less change impact on the affected cluster, so does it for the non-target module. Then the SDCP is defined as:

$$P = w_{TC} \times S_{TC} + w_{AC} \times S_{AC} + w_{NC} \times S_{NC}, \tag{8}$$

where w_{TC}, w_{AC}, and w_{NC} represent the affected degree of these three clusters caused by design change after



	1	2	3	4	5
1	1	0.4	0.2		
2		1			0.2
3	0.5		1		
4				1	0.8
5					1

(*a*) low strength of interface

	1	2	3	4	5
1	1	0.4	0.2		
2		1			0.6
3	0.5		1		
4				1	0.8
5					1

(b) high strength of interface

	1	2	3	4	5
1	1	0.4	0.2	0.6	
2		1			0.2
3			1		
4				1	0.8
5					1

(*c*) two interfaces

Fig. 6 Three kinds of interfaces

reducing the relationships of the interfaces. S_{TC}, S_{AC}, S_{NC} represent the scope of design change in the target cluster, the affected cluster, and the non-target cluster, respectively.

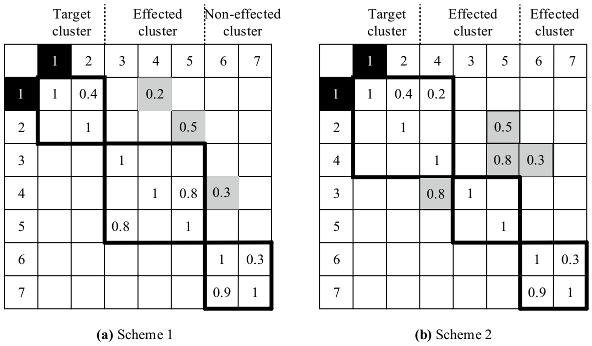
4. Modified SDCP considering the punishment of interfaces

According to Eq. (8), the SDCP of both Fig. 6a, b can be expressed as $P=(4.1+3\times 0.05)\times w_{\mathrm{TC}}+(2.8+1\times 0.05)\times w_{\mathrm{AC}}$. However, for the same interface, if the strength of the interface is higher, then it is much harder to modify. And if the same measure to modify the interface, the interface with high strength is more likely to propagate the change. At the same time, as shown in Fig. 6a, c, with the increase of the number of interfaces, the design change will be easier to propagate to other clusters. Thus, to reasonably define the SDCP, the interfaces are taken as the punishment elements which should be as few as possible and with a small strength. Then the modified SDCP can be calculated as follows.

$$\mathrm{MP} \;=\; w_{\mathrm{TC}}\times S_{\mathrm{TC}}+w_{\mathrm{AC}}\times S_{\mathrm{AC}}+w_{\mathrm{NC}}\times S_{\mathrm{NC}}+w_{\mathrm{I}}\times I, \tag{9}$$

where w_{I} is the penalty factor for the interfaces, I is the summation of the strength for all the interfaces between modules, w_{TC}, w_{AC}, and w_{NC} refer to the value obtained by Li and Chen (2014), and w_{I} is determined after discussion with the designers. Then, the four parameters are normalized as $w_{\mathrm{TC}}=0.3$, $w_{\mathrm{AC}}=0.2$, $w_{\mathrm{NC}}=0.1$, $w_{\mathrm{I}}=0.4$. The design change is not a deterministic process, but also a matter of choice or external constraints. In this study, the modified SDCP is selected as one of the two optimization objectives to establish the module partition, which means that, the rationality of the modularization is termed as a primary constraint for the design change propagation.

As shown in Fig. 7, there are two module partition schemes, and $\mathrm{MP}_1=(2.4+0.05)\times 0.3+(4.6+4\times 0.05)\times 0.2+3.2\times 0.1+(0.2+0.5+0.3)\times 0.4=2.415$, $MP_2=(3.6+4\times 0.05)\times 0.3+(2+2\times 0.05)\times 0.2+3.2\times 0.2+(0.5+0.8+0.3+0.8)\times 0.4=3.16$, respectively. Obviously, $MP_1<MP_2$, this is because in scheme 1, the highly connected components are sorted into the same cluster and only fewer interfaces exist.



	1	2	3	4	5	6	7
1	1	0.4		0.2			
2		1			0.5		
3			1				
4				1	0.8	0.3	
5			0.8		1		
6						1	0.3
7						0.9	1

(a) Scheme 1

	1	2	4	3	5	6	7
1	1	0.4	0.2				
2		1			0.5		
4			1		0.8	0.3	
3			0.8	1			
5					1		
6						1	0.3
7						0.9	1

(b) Scheme 2

Fig. 7 Illustration of two module partition schemes

3.3 A bi-objective optimization model for modularization

In the modularization of complex products, the modularity that is a benefit criterion should be as large as possible. On the contrary, the modified SDCP should be minimized since it is a cost criterion. Then the optimization model is constructed as follows.

$$F(X)=\left\{\max\left(\sum_{i,j}[L_{ij}-\pi_i\pi_j]\delta(c_i,\ c_j)\right),\right.$$
$$\left.\min\left(w_{\mathrm{TC}}\times S_{\mathrm{TC}}+w_{\mathrm{AC}}\times S_{\mathrm{AC}}+w_{\mathrm{NC}}\times S_{\mathrm{NC}}+w_{\mathrm{I}}\times I\right)\right\}, \tag{10a}$$

$$\text{s.t.}\ \ 0\le\sum_{i,j}[L_{ij}-\pi_i\pi_j]\delta(c_i,\ c_j)\le 1, \tag{10b}$$

$$0<w_{\mathrm{TC}}\times S_{\mathrm{TC}}+w_{\mathrm{AC}}\times S_{\mathrm{AC}}+w_{\mathrm{NC}}\times S_{\mathrm{NC}}+w_{\mathrm{I}}\times I<\frac{2}{5}n^2, \tag{10c}$$

where the max $\left(\sum_{i,j}[L_{ij}-\pi_i\pi_j]\delta(c_i,\ c_j)\right)$ is used to maximize the modularity that ranges from 0 to 1; the min $\left(w_{\mathrm{TC}}\times S_{\mathrm{TC}}+w_{\mathrm{AC}}\times S_{\mathrm{AC}}+w_{\mathrm{NC}}\times S_{\mathrm{NC}}+w_{\mathrm{I}}\times I\right)$ is established to minimize the modified SDCP. Because the interfaces occupy the greatest weight in the evaluation of the modified SDCP, when all relationships between components are regarded as the interfaces, the modified SDCP reaches the upper limit $\frac{2}{5}n^2$.

3.4 A solving algorithm based on NSGA-II

The bi-objective model for modularization is a typical combinatorial optimization problem, which is intractable to be solved in a reasonable time by traditional mathematical methods. With the in-depth study of the bi-objective problem, the bi-objective genetic algorithm is proved to have a good performance in handling such problems, especially the NSGA-II algorithm proposed by Deb et al. (2002). The NSGA-II algorithm develops the fast-non-dominated sorting and elite selection strategy after improving the NSGA and uses the congestion instead of the fitness value sharing, which effectively overcomes the problems of low robustness and long running time of the traditional bi-objective algorithm. Therefore, the NSGA-II is employed to solve the pre-established bi-objective modularization model incorporating modularity and modified SDCP. The flow of the NSGA-II is shown in Fig. 8.

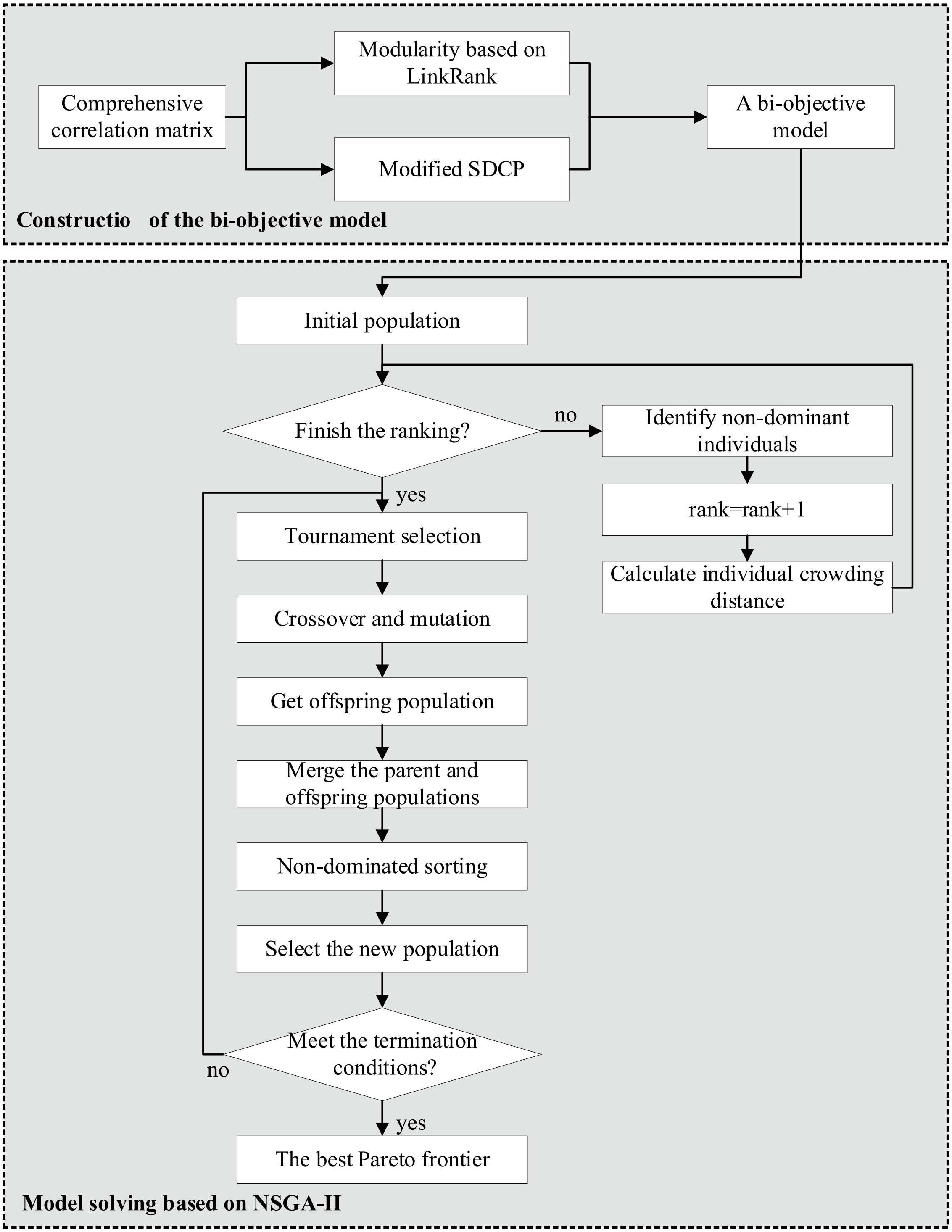

Comprehensive correlation matrix
Modularity based on LinkRank
Modified SDCP
A bi-objective model
Constructio of the bi-objective model
Initial population
Finish the ranking?
no
Identify non-dominant individuals
rank=rank+1
Calculate individual crowding distance
yes
Tournament selection
Crossover and mutation
Get offspring population
Merge the parent and offspring populations
Non-dominated sorting
Select the new population
Meet the termination conditions?
no
yes
The best Pareto frontier
Model solving based on NSGA-II

Fig. 8 Solving procedure of the NSGA-II

Components	1	2	3	4	5	6	7	8
Modules	1	3	2	1	1	4	2	3

Fig. 9 One-dimensional real number coding

To generate the initial population, a one-dimensional real number coding method is addressed in this study, as shown in Fig. 9. This method marks a module index for each component, thus it only needs an $O(n)$ storage space for a model

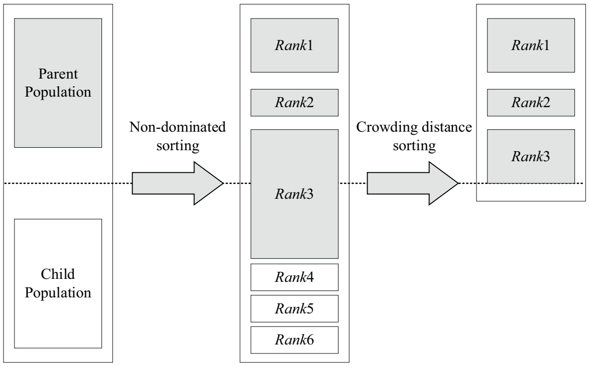


Fig. 10 Non-dominated sorting and elite selection

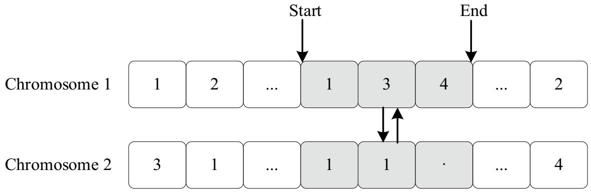


Fig. 11 Crossover

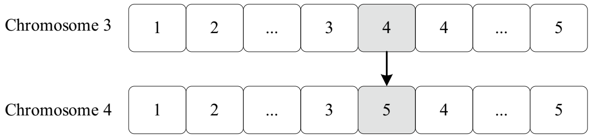


Fig. 12 Mutation

with n scale, which is smaller than the 0–1 matrix coding method.

The non-dominated sorting and elite selection are the critical idea of NSGA-II. As shown in Fig. 10, the non-dominated sorting firstly classifies each component according to its dominance degree and then sorts the components at the same level according to their crowding distances. The elite selection is to sort all individuals by merging parent and child populations and select the top-ranked individuals that meet the population size.

The partial-mapped crossover is employed in this study. The starting and ending positions are randomly selected from the parent individuals as shown in Fig. 11. As to the mutation, the module number of a component in a chromosome is changed to another one with a certain probability. Figure 12 shows the one-point mutation of chromosomes.

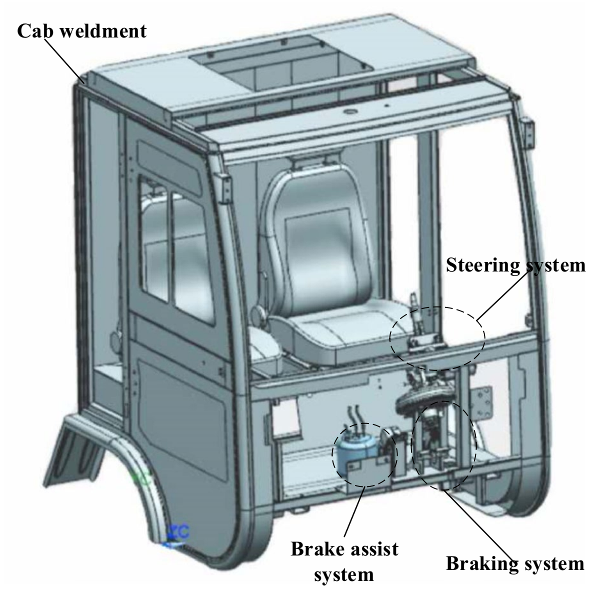


Fig. 13 Cab of the electric sanitation vehicles

4 Case study

With the continuous strengthening of environmental protection, the demand for electric sanitation vehicles is increasing. Due to dynamic internal and external factors, the design changes of electric sanitation vehicles are inevitable. As the key subsystem of the electric sanitation vehicle, the cab is easily affected by dynamic design changes, such as a more comfortable environment and more sensitive braking performance. Thus, the modularization of the cab considering modified SDCP is implemented as an example to verify the feasibility of the proposed method.

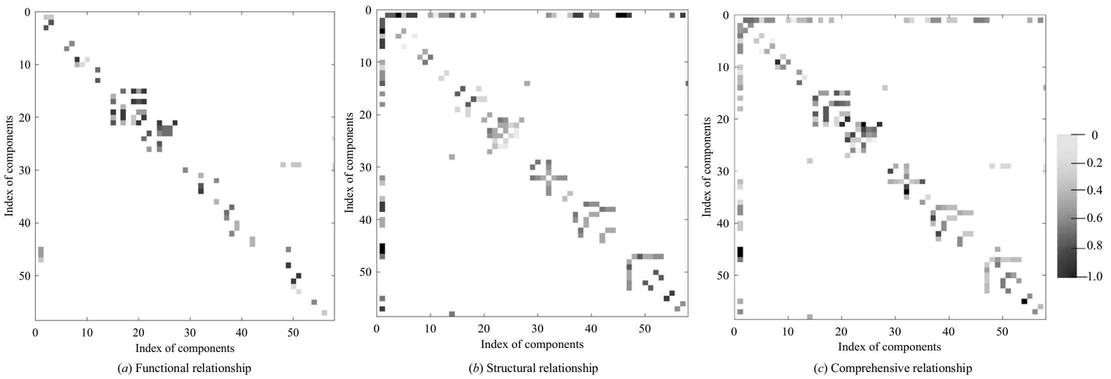
4.1 Quantification of component relationship and elements identification

As shown in Fig. 13, the cab is mainly composed of the cab weldment, brake system, brake auxiliary system, steering system, which includes 58 components (Table 3). Then the functional and structural relationship (see Fig. 14a, b) are quantified according to the evaluation criteria in Tables 1 and 2. After discussing with the designers, setting $w_f = w_s = 0.5$. According to Eq. (1), the CCM of cab components is established as shown in Fig. 14c.

According to the influence of components in the product structure and their change probability, the brake master cylinder (21) is selected as the TE, and then the AEs and NEs are identified as shown in Table 4.

Table 3 Components of the cab

No.	Component	No.	Component	No.	Component
1	Cab weldment	21	Brake cylinder	40	Spring bracket 1
2	Wheel eyebrow 1	22	Brake shaft	41	Spring bracket 2
3	Wheel eyebrow 2	23	Bracket weldment	42	Right gas spring
4	Corner aluminum	24	Pedal lever	43	Spring bracket 3
5	Blue leather	25	Support spring	44	Spring bracket 4
6	Cab pedal	26	Composite sleeve	45	Front windshield
7	Aluminum plate	27	Oil pot	46	Glass behind door
8	Hand brake	28	Rotary knob	47	Front face
9	Brake bracket	29	Combination switch	48	Wiper motor
10	Brake box weldment	30	Direction lock	49	Wiper
11	Shock absorber strap	31	Decorative cover	50	Left headlight
12	Seat	32	Steering column	51	Right headlight
13	Self-locking hasp	33	Direction controller	52	Plate 1
14	Instrument desk	34	Steering wheel	53	Plate 2
15	Booster motor	35	Universal joint coupling	54	Backward mirror
16	Motor bracket	36	Lower shaft dust jacket	55	Mirror bracket
17	Vacuum pump	37	Left door assembly	56	Roof lamp
18	Pump bracket	38	Right door assembly	57	Mounting plate
19	Trachea 1	39	Left gas spring	58	Instrument
20	Trachea 2				



Relationship of components: **(a)**Functional relationship **(b)**Structural relationship **(c)**

Fig. 14 Relationship of components: **a** Functional relationship, **b** Structural relationship, **c** comprehensive relationship

Table 4 Result of the elements identification

Category	Components number
TE	21
AE	15, 17, 19, 23, 24, 27
NE	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 18, 22, 25, 26, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58

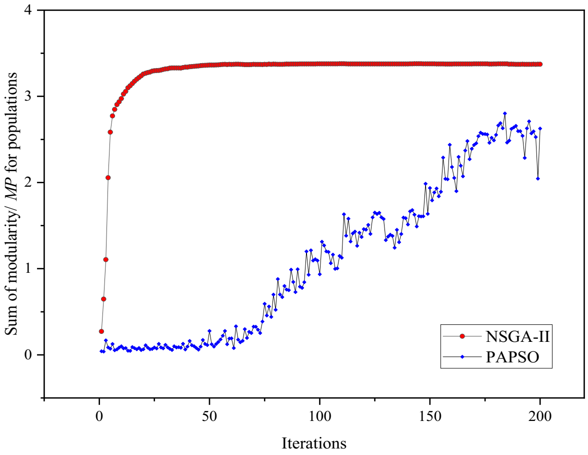


Fig. 15 Convergence process of the NSGA-II and PAPSO

4.2 Modularization based on NSGA-II

The NSGA-II is implemented to solve the bi-objectives model. The algorithm parameters are set as: the initial population is 100, the number of generations is 200, the crossover probability is 0.9, and the mutation probability is 0.1. The algorithm is developed by Matlab 2012, and the computer is configured with inter (R) Core TM i5-5200U CPU (2.20 GHz) and 4 GB RAM.

As shown in Fig. 15, after 80 iterations, the sum of the quotient of the modularity and modified SDCP in the population reaches a stable state, which shows that the algorithm has good convergence. After 200 iterations, there are 16 Pareto frontier solutions, as shown in Fig. 16a. The modularity and modified SDCP corresponding to each scheme are listed in Table 5.

Choosing scheme 5 as an example, it has the same components in M_1, M_2, M_6, and M_7 compared with the existing scheme as shown in Table 6. That is, the modularization scheme considering modified SDCP is consistent with part of the existing modules. Meanwhile, as shown in Fig. 17a, b, when component 21 changes, the optimal change path obtained by Li et al. (2019a) is 21 → 20 → 15 → 16 → 1. In scheme 5, two modules (M_3 and M_5) are affected by this change. However, three modules (M_3, $M_{8,}$ and M_{10}) are affected in the existing modular structure. After considering the modified SDCP, components with high change probability are increasingly easier to be divided into the same module. So the design change can be controlled in few modules, which effectively reduce the change impact and improve the agility of complex product design.

Table 5 Optimal scheme and target value

Scheme	Modularity	MP	Scheme	Modularity	MP
1	0.7929	22.470	9	0.7433	21.915
2	0.7912	22.440	10	0.7406	21.905
3	0.7780	22.245	11	0.7106	20.670
4	0.7529	22.080	12	0.7101	20.620
5	0.7349	21.585	13	0.7968	22.880
6	0.7332	21.555	14	0.7969	22.878
7	0.7021	20.530	15	0.8052	23.350
8	0.7996	22.890	16	0.7185	20.770

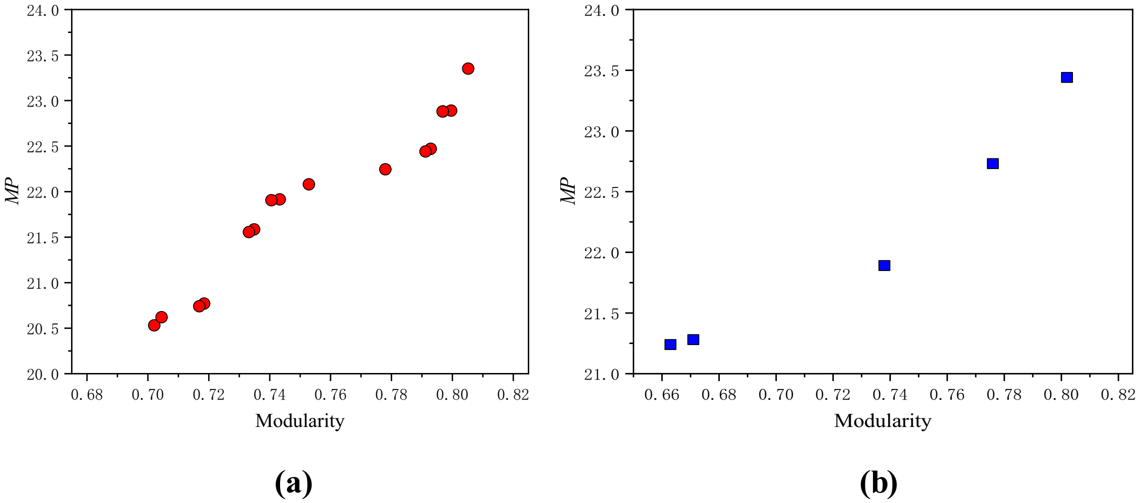
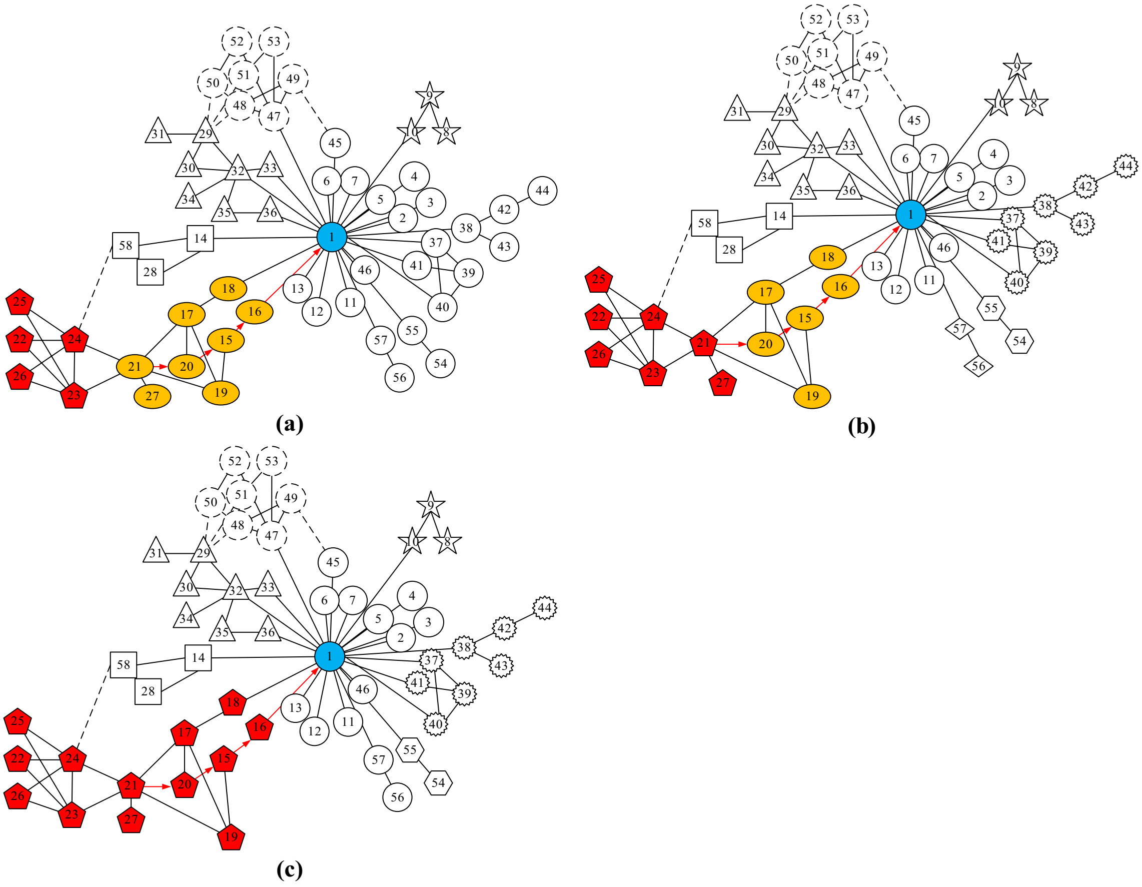


Fig. 16 Pareto frontier solution: a Pareto frontier solution of NSGA-II, b Pareto frontier solution of PAPSO

Table 6 Three modularization schemes

No	Modules of scheme 5	Existing modules	Modules of the traditional method
M_1	8, 9, 10	8, 9, 10	8, 9, 10
M_2	29, 30, 31, 32, 33, 34, 35, 36	29, 30, 31, 32, 33, 34, 35, 36	29, 30, 31, 32, 33, 34, 35, 36
M_3	1, 2, 3, 4, 5, 6, 7, 11, 12, 13, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 54, 55, 56, 57	1, 2, 3, 4, 5, 6, 7, 11, 12, 13, 45, 46	1, 2, 3, 4, 5, 6, 7, 11, 12, 13, 45, 46, 56, 57
M_4	22, 23, 24, 25, 26	54, 55	15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27
M_5	15, 16, 17, 18, 19, 20, 21, 27	56, 57	54, 55
M_6	14, 28, 58	14, 28, 58	14, 28, 58
M_7	47, 48, 49, 50, 51, 52, 53	47, 48, 49, 50, 51, 52, 53	47, 48, 49, 50, 51, 52, 53
M_8	–	15, 16, 17, 18, 19, 20	37, 38, 39, 40, 41, 42, 43, 44
M_9	–	37, 38, 39, 40, 41, 42, 43, 44	–
M_{10}	–	21, 22, 23, 24, 25, 26, 27	–
Modularity	0.7349	0.7525	0.8033
MP	21.585	22.651	23.362



(a)

(b)

(c)

Fig. 17 **a** Modules of scheme 5, **b** existing modules, **c** modules without considering modified SDCP

Table 7 Comparison with NAGA-II and PAPSO

Options	NSGA-II	PAPSO
Convergence	Fast	Slow
Running time	36.13 s	38.67 s
Number of optimal solutions	18	5
Distribution	Excellent	Good

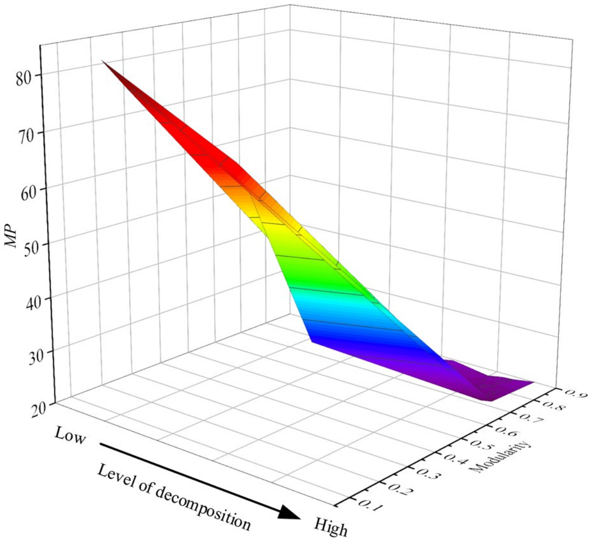


Fig. 18 Sensitivity analysis for the impact on modularity and modified SDCP by the level of decomposition

4.3 Discussion

4.3.1 Comparison with the modularization without considering the design change propagation

The traditional modularization methods usually divide modules based on the static relationship between components without considering the impact of design change propagation. This study compares the modularization method considering modified SDCP with the traditional modularization method mentioned by Zhang et al. (2019). The results are shown in Table 6 and Fig. 17a, c. Although the modularity of scheme 5 obtained by the proposed approach is a little smaller than that obtained by Zhang et al. (2019), the modified SDCP of scheme 5 performs better. This means the proposed method could obtain a modular structure that can control the design change propagation as well as keep the modularity. As for the number of affected modules by the above design change path, both two schemes have two affected modules. However, the number of components in the affected modules of scheme 5 is fewer.

4.3.2 Comparison with the Pareto archive particle swarm optimization (PAPSO) algorithm

The proposed NSGA-II is compared to the Pareto archive particle swarm optimization (PAPSO) algorithm (Wei et al. 2018) which also is a commonly applied algorithm to solve the multi-objective optimization model. The two algorithms are compared against a variety of different metrics [i.e., the convergence of algorithms (Fig. 15), running time, number of optimal solutions, and covering space of Pareto frontier (Fig. 16)]. And the comparison results are summarized in Table 7. In particular, the NSGA-II significantly outperformed the PAPSO algorithm concerning the four metrics.

As shown in Fig. 15, the NSGA-II algorithm reaches convergent at the 80th iteration, while the PAPSO algorithm has not converged until the end of the programming. And as shown in Fig. 16, the NSGA-II gets 16 optimal frontier solutions, while the PAPSO algorithm gets only 5 optimal frontier solutions. And the solution in the interval of 0.66 to 0.68 based on the PAPSO algorithm is smaller compared with the solutions in the interval of 0.70–0.72 by the NSGA-II. At the same time, the solutions obtained based on the NSGA-II are distributed in every interval, while the solutions based on the PAPSO algorithm do not exist in the interval of 0.68–0.72. Thus, the Pareto frontier obtained by the NSGA-II is better than those of the PAPSO algorithm in distribution.

4.3.3 Sensitivity analysis for the level of decomposition

As shown in Fig. 18, the different levels of decomposition for the product will affect the modularity and modified SDCP for the module partition schemes. With the increasing level of product decomposition, the modularity of the scheme for modular design is increasing and the modified SDCP reduces gradually. This is because as the degree of decomposition increases, much more components can be acquired. And the components that have a high relationship with each other will be integrated into a module that has an independent function. Besides, the interface between modules can be identified and modified to control the design change propagation.

5 Conclusion

Change propagation is inevitable in complex product design. And there is a conspicuous contradiction between the control of design change propagation and product modularization. To ensure modularity as well as effectively control the change propagation, a bi-objective model incorporating modularity and modified SDCP is developed, which considers the impact of design change propagation in the modularization for complex products. The main conclusions are concluded as follows.

(1) Due to the lack of consideration for design change propagation in the existing modularization method, this study constructs a bi-objective model considering modularity and modified SDCP, and solves the model by the NSGA-II. The obtained modularization schemes can effectively cope with the dynamic change of products while maintaining reasonable modularity.
(2) To determine the size of change propagation and effectively control the design change, a modified SDCP that considers the punishment of the interface is proposed, through which the scope of change propagation can be quantified, and the change propagation impact can be controlled by focusing on the key interfaces.
(3) Taking the modular design of the electric sanitation car cab as an example, the validity and feasibility of the proposed method are verified.

In the future study, the influence of the existing modular structure on the control of design change propagation in complex product design will be studied, especially focusing on the penalty mechanism when design changes are transmitted to components in different modules.

Acknowledgements This project was supported by the National Natural Science Foundation, China (nos. 51505480, 72001203, 51875345).

References

Algeddawy T, Elmaraghy H (2013) Optimum granularity level of modular product design architecture. CIRP Ann-Manuf Technol 62:151–154. https://doi.org/10.1016/j.cirp.2013.03.118

Beek V, Thom J, Erden M (2010) Modular design of mechatronic systems with function modeling. Mechatronics 20:850–863. https://doi.org/10.1016/j.mechatronics.2010.02.002

Braha D (2016) The complexity of design networks: Structure and dynamics. Experimental design research. Springer, Cham, pp 129–151

Braha D, Bar-Yam Y (2004a) Topology of large-scale engineering problem-solving networks. Phys Rev E 69:016113. https://doi.org/10.1103/PhysRevE.69.016113

Braha D, Bar-Yam Y (2004b) Information flow structure in large-scale product development organizational networks. J Inf Technol 19(4):244–253. https://doi.org/10.1057/palgrave.jit.2000030

Braha D, Bar-Yam Y (2007) The statistical mechanics of complex product development: empirical and analytical results. Manage Sci 53(7):1127–1145. https://doi.org/10.1287/mnsc.1060.0617

Braha D, Maimon O (1998) A mathematical theory of design: foundations, algorithms and applications, vol 17. Springer Science & Business Media, New York

Braha D, Yassine A (2003) Complex concurrent engineering and the design structure matrix method. Concurrent Eng-Res A 11:165–176. https://doi.org/10.1177/106329303034503

Brahma A, Wynn DC (2020) Margin value method for engineering design improvement. Res Eng Des 31:353–381. https://doi.org/10.1007/s00163-020-00335-8

Brin S, Page L (2012) Reprint of: anatomy of a large-scale hypertextual web search engine. Comput Netw 56:3825–3833. https://doi.org/10.1016/j.comnet.2012.10.007

Cheng X, Li J, Wan C, Qiu H, Wan Y (2018) Dbscan-based modular design for the crane grab. Int J Wirel Mob Comput 15:157–162. https://doi.org/10.1504/IJWMC.2018.095693

Clarkson PJ, Simons C, Eckert C (2014) Predicting change propagation in complex design. J Mech Des 126:788–797. https://doi.org/10.1115/1.4027495

Cohen T, Navathe SB, Fulton RE (2000) C-FAR, change favorable representation. Comput Aided Des 32:321–338. https://doi.org/10.1016/S0010-4485(00)00015-4

Deb K, Prata A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6:182–197. https://doi.org/10.1109/4235.996017

Eckert C, Clarkson PJ, Zanker W (2004) Change and customization in complex engineering domains. Res Eng Des 15:1–21. https://doi.org/10.1007/s00163-003-0031-7

Eckert C, Isaksson O, Lebjioui S, Earl C, Edlund S (2020) Design margins in industrial practice. Des Sci 6:E30. https://doi.org/10.1017/dsj.2020.19

Gershenson JK, Prasad GJ, Zhang Y (2003) Product modularity: definitions and benefits. J Eng Des 14:295–313. https://doi.org/10.1080/0954482031000091068

Guo F, Gershenson JK (2007) Discovering relationships between modularity and cost. J Intell Manuf 18:143–157. https://doi.org/10.1007/s10845-007-0007-y

Jarratt T, Eckert C, Caldwell N, Clarkson PJ (2011) Engineering change: an overview and perspective on the literature. Res Eng Des 22:103–124. https://doi.org/10.1007/s00163-010-0097-y

Ji Y, Jiao RJ, Chen L, Wu C (2013) Green modular design for material efficiency: a leader-follower joint optimization model. J Clean Prod 41:187–201. https://doi.org/10.1016/j.jclepro.2012.09.022

Kim Y, Son SW, Jeong H (2010) Finding communities in directed networks. Phys Rev E 81:016103. https://doi.org/10.1103/PhysRevE.81.016103

Li S, Chen L (2014) Identification of clusters and interfaces for supporting the implementation of change requests. IEEE Trans Eng Manage 61:323–335. https://doi.org/10.1109/TEM.2013.2292856

Li J, Zhang H, Gonzalez MA, Yu S (2008) A multi-objective fuzzy graph approach for modular formulation considering end-of-life issues. Int J Prod Res 46:4011–4033. https://doi.org/10.1080/00207540601050376

Li Y, Chu X, Chu D, Liu Q (2014) An integrated module partition approach for complex products and systems based on weighted complex networks. Int J Prod Res 52:4608–4622. https://doi.org/10.1080/00207543.2013.879617

Li Q, Efatmaneshnik M, Ryan M, Shoval S (2017a) Product modular analysis with design structure matrix using a hybrid approach based on MDS and clustering. J Eng Des 28:433–456. https://doi.org/10.1080/09544828.2017.1325858

Li Y, Wang Z, Zhang L, Chu X, Xue D (2017b) Function module partition for complex products and systems based on weighted and directed complex networks. J Mech Des 139:1–13. https://doi.org/10.1115/1.4035054

Li Y, Li M, Wang Z (2019a) Multi-source design change propagation path optimization for complex product based on weighted and directed network model. J Mech Eng 55:213–222. https://doi.org/10.3901/JME.2019.06.213

Li Z, Wang S, Yin W (2019b) Determining optimal granularity level of modular product with hierarchical clustering and modularity assessment. J Braz Soc Mech Sci 41:342–355. https://doi.org/10.1007/s40430-019-1848-y

Martin MV, Ishii K (2002) Design for variety: developing standardized and modularized product platform architectures. Res Eng Des 13:213–235. https://doi.org/10.1007/s00163-002-0020-2

Morkos B, Shankar P, Summers JD (2012) Predicting requirement change propagation, using higher order design structure matrices: an industry case study. J Eng Des 23:905–926. https://doi.org/10.1080/09544828.2012.662273

Newman ME (2004) Analysis of weighted networks. Phys Rev E 70:056131. https://doi.org/10.1103/PhysRevE.70.056131

Park K, Kremer G (2019) An investigation on the network topology of an evolving product family structure and its robustness and complexity. Res Eng Des 30:381–404. https://doi.org/10.1007/s00163-019-00310-y

Prencipe A (2000) Breadth and depth of technological capabilities in CoPS: the case of the aircraft engine control system. Res Policy 29:895–911. https://doi.org/10.1016/S0048-7333(00)00111-6

Salvador F, Villena VH (2013) Supplier integration and NPD outcomes: conditional moderation effects of modular design competence. J Supply Chain Manag 49:87–113. https://doi.org/10.1111/j.1745-493x.2012.03275.x

Sinha K, Han S, Suh ES (2020) Design structure matrix-based modularization approach for complex systems with multiple design constraints. Syst Eng 23:211–220. https://doi.org/10.1002/sys.21518

Ullah I, Tang D, Yin L, Hussain I, Wang Q (2018) Cost-effective propagation paths for multiple change requirements in the product design. Proc Inst Mech Eng Part C-J Eng Mech Eng Sci 232:1572–1585. https://doi.org/10.1177/0954406217707788

Wei W, Liang H, Wuest T, Liu A (2018) A new module partition method based on the criterion and noise functions of robust design. Int J Adv Manuf Tech 94:3275–3285. https://doi.org/10.1007/s00170-016-9797-4

Yassine A, Joglekar N, Braha D, Eppinger S, Whitney D (2003) Information hiding in product development: the design churn effect. Res Eng Des 14:145–161. https://doi.org/10.1007/s00163-003-0036-2

Yu T, Yassine A, Goldberg D (2007) An information theoretic method for developing modular architectures using genetic algorithms. Res Eng Des 18:91–109. https://doi.org/10.1007/s00163-007-0030-1

Yu S, Yang Q, Jing T, Tian X, Yin F (2011) Product modular design incorporating life cycle issues—Group Genetic Algorithm (GGA) based method. J Clean Prod 19:1016–1032. https://doi.org/10.1016/j.jclepro.2011.02.006

Zhang N, Yang Y, Zheng Y, Su J (2019) Module partition of complex mechanical products based on weighted complex networks. J Intell Manuf 30:1973–1998. https://doi.org/10.1007/s10845-017-1367-6