Hindawi
Advances in Mathematical Physics
Volume 2022, Article ID 5158114, 19 pages
https://doi.org/10.1155/2022/5158114

Research Article

Research on Customer Requirement-Driven Individualized Product Module Division and Configuration Based on Community Structure

Qin Yang , Xuzheng Li, Qunqun Yin, Fan Liu, and Weijie Chang

School of Mechanical Engineering, Hefei University of Technology, Hefei 230009, China

Correspondence should be addressed to Qin Yang; yangqin@hfut.edu.cn

Received 23 March 2022; Revised 4 July 2022; Accepted 13 July 2022; Published 3 August 2022

Academic Editor: Sayyouri Mhamed

The main goal of designing individualized products is to meet the specific requirements presented by the customer. Therefore, designers need to adapt the relevant components of the product for each customer, which is difficult to achieve efficiently with existing methods. In this paper, we propose an integrated approach that enables intuitive modelling of products and supports fast conversion of different customer requirements (CRs) to configuration solutions. Firstly, the product properties are decomposed, and a standardized CR expression template is established to enable the mapping of CR information to product properties change information. Secondly, the community structure in complex network theory is introduced to visualize and quantitatively describe the relationship between product parts. The community detection method based on fuzzy clustering is applied to module division, and the optimal result is obtained using the F-statistic and modularity as evaluation indexes. Finally, the individualized product configuration problem is transformed into a dynamic constraint satisfaction problem. According to CR information and product module division result, the hierarchical community structure is used to narrow the search space and quickly derive solutions with high customer satisfaction through case-based configuration. An automatic guided vehicle is used as an example to illustrate the effectiveness and practicality of this approach.

1. Introduction

Today's consumers are increasingly eager to design and manufacture individualized products that reflect their specific needs [1]. In the upcoming paradigm of mass individualized, customers will initiate the product and manufacturers will produce at low cost [2]. Many scholars have proposed methods that consider customer requirements (CRs) in the past three decades. The design for variety method proposed by Martin and Ishii can identify components that need to be redesigned because of changing requirements. Krause et al. [3] developed a multicriterion approach using the variety allocation model to relate different CRs to variant components. Blee et al. [4] proposed an integration method based on variety and modular design to reduce the scope of variant components. Otto et al. [5] link different branches of product design research into a logical sequences that can actually be used for product platform development. Simpson et al. [6] introduced an approach to integrate several tools into a framework to translate CRs into commonality specifications. Hu et al. [7] envisaged that the product should contain three types of modules: normal modules, customized modules, and individualized modules. They all have standard interfaces to allow for easy assembly and disassembly. To conclude, individualized product design requires not only an accurate identification of CRs but also a rational architecture that can describe the structure of their functions and how they are implemented.

Generally speaking, CRs are dynamic and customers are not aware of the technical characteristics of the product but only focus on product behaviour they cause [8–10]. Therefore, tools such as Voice of Customer, KANO-model, Quality Function Deployment, Preference Graph, Kansei engineering, and Bayesian networks, are proposed to collect and analyse CRs [11–16]. These tools can be very effective in mapping requirements into product design parameters but still have shortcomings. In the individualized product market, market expectations are characterised by CRs. If the analysis of requirements is done by designers, it is impossible to include every customer's needs. On the contrary, if it is done by the customers themselves, who do not have the expertise, this would be too difficult. Therefore, an approach is needed to simplify the analysis of CRs.

In the area of individualization, adjustments to the product components are allowed. But also, here, ideally, the product structure remains unchanged between modular product families. Grouping components with similar characteristics can improve product maintainability and have a significant impact on the overall life cycle [17, 18]. Stone et al. [19] provided a unified description of product functions and proposed a three-stage heuristic algorithm to construct a modular structure. Kusiak and Huang [20] used graph theory as a tool to express the physical connection relationship between parts and tried to explore the balance between product performance and cost using a fuzzy neural network method. Marshall and Leaney [21] developed a new modular design approach by introducing the concept of Holon in systems engineering. Pimmler and Eppinger [22] decomposed products into different components and then cluster them into chunks by design structure matrix, taking into account spatial, energy, information, and materials. Erixon [23] took product strategy considerations into modularization and proposed the Modular Function Deployment method based on Module Indication Matrix, which links the components of a product with 12 modular drivers such as design development, differentiation, and manufacturing. Individualized products may have many parts, and using the previously proposed method may result in a huge workload. More importantly, they are not intuitive to the customer due to the lack of visualization of the product structure.

Product configuration has received continued attention from academia and industry as a technology that can deliver customized products to customers in a short period of time [24, 25]. The current product configuration approaches can be classified as rule-based, model-based, and case-based [26]. Each of the three approaches has advantages and disadvantages. The most typical model of rule-based configuration is DEC's computer configuration system R1/XCON, which has roughly 31,000 parts and 17,500 configuration rules in its database [27]. In the early days, this technique was very popular, but when the rules changed, a large amount of knowledge in the database needed to be modified, which led to a heavy workload. Huang et al. [28] represented the product information, constraints, configuration rules, and their mapping relationships as a semantic model, which improved the flexibility of rule-based configuration. Model-based configuration usually requires the construction of a good generic model of the product, which is subsequently transformed into a constraint satisfaction problem (CSP) [29–32]. Jiao et al. [33] proposed an integrated model for market-based product transaction data to generating feasible configuration plans and selecting the best configuration solution based on CRs. The model-based configuration is more comprehensive in considering factors, but lacks generality in solving process. The case-based configuration is to find the product with the highest similarity to the CRs in the enterprise's product database [34–36]. This method is highly maintainable for configuration models, but when the configuration knowledge is not perfect, the obtained solution is not necessarily optimal, and when there are many existing cases, the retrieval process is time-consuming, which can seriously affect the configuration efficiency. Wang et al. [37] proposed a case-based method with self-organizing mapping and fuzzy similarity priority comparison to narrow the search scope, thus improving the retrieval efficiency.

Two difficulties and challenges were identified by reviewing the results of research on CR modelling and product modelling.

(1) Customer involvement in the product design process brings with it a lot of uncertainty. The whole process needs to consider the customers' user experience. In order to facilitate the customer, it is important to dynamically describe CRs and relate them directly to individual product variant components

(2) As the number of parts increases, it will become more difficult to model, display, and configure individual products. The efficiency of the existing methods will decrease significantly

Properties describe the product's behaviour that cannot be directly influenced by the designer, e.g., function, weight, safety, and reliability [38]. The term product attribute refers to a specific design of a product property such as a weight of 200 kg or 300 kg [39]. By predefining product components corresponding to different attributes and then combining them, it is possible to meet almost any specific CR.

Complex network theory has been widely used in supply chains, electric power, and urban transportation [40–42]. Newman and Girvan [43] proposed the concept of community structure in complex networks, but there is still no uniform definition. In general, it is believed that the relationship between nodes within a community is tight and the nodes between the communities are loose. Because of the similarities between the community structure and the modular topology of the product, complex networks can represent the product visually. Fan [44] applied it to model the product family structure, but they only considered the physical relationships between components, which is not consistent with engineering reality. In addition, the drawback of the case-based configuration approach is the inefficiency of the retrieval due to the increasing number of product parts. By introducing the concept of hierarchical community structure and narrowing the scope of the search, product configuration time can be reduced.

Based on the above mentioned, this paper proposes an integrated approach whose goal is to quickly transform CRs into an individualized product model. The method allows to adapt the product components for each customer without taking a lot of time. Firstly, product properties are decomposed and classified. A standard CR template is established to facilitate the mapping of requirement information to product properties change information. Secondly, the community structure in complex network theory is applied to the representation of individualized products. Physical correlation matrix, functional correlation matrix, and structural correlation matrix among product parts are established. Subsequently, all module division schemes are derived by a community detection method based on fuzzy clustering. The F-statistic method is used to preselect the schemes to avoid wasting unnecessary resources, and the optimal division scheme is obtained by using the evaluation of the modularity of the community. Finally, combining the advantages of model-based and case-based product configuration methods, the individualized product configuration problem (PCP) is transformed into a dynamic constraint satisfaction problem (DCSP). Hierarchical constraints of complex variables are established by the introduced complex hierarchical community structure to reduce the search scope of individualized product components, so that the solution of configuration can be obtained quickly. This approach not only shortens the product development cycle but also provides customers with an intuitive configuration process.

The rest of the paper is organized as follows. The properties of the individualized products are decomposed from the perspective of CRs in Section 2. Section 3 applies complex network theory to the modularization of individualized products. Section 4 transforms the individualized PCP into DCSP and solves it quickly by case-based configuration method. Section 5 uses an example to show that the approach is effective and reasonable. The results of the example are discussed in Section 6. Conclusions and future works are presented in Section 7.

2. Property Decomposition of Individualized Product

CRs are often specific to product functionality, performance, or other aspects. In order to correlate CRs with individual product components, an effective product decomposition is established. The product is decomposed at the property level, subsequently clarifying the different types of properties, the attributes contained in the properties, and the components corresponding to the properties.

2.1. Concept of Properties. For an individualized product, it usually has many different properties. Each property can be described in a different way. For example, automated guided vehicle (AGV) has appearance property, moving property, power property, etc. Among them, the moving attribute can be described by moving mode, moving speed, moving stability, etc. By decomposing the product properties, all aspects of a product can be described. However, the results may vary due to the different criteria of decomposition. In this paper, the decomposition of product properties is expressed as

$$P = \{c_1, c_2, c_3, \cdots, c_n\}, \tag{1}$$

where P denotes the product and c_n denotes the n-th property of the product after decomposition.

To decompose products from the perspective of properties, on the one hand, a detailed analysis of the product is required to ensure a full description of the product. On the other hand, it should be based on historical data of the company's interactions with customers to decompose property information that customers can understand intuitively.

2.2. Relationship between Properties and Components. In this paper, components are defined as the parts that constitute the function or performance of a product, and these parts are combined together by certain principles to determine a certain property of the product. The properties cannot be changed directly. If a property needs to be changed, it must be realized by changing the components that make up the property; i.e., the function and performance of the product must be changed by changing the components of the product. The relationship between properties and components is shown in Figure 1.

The existence of a binding relationship between properties depends on whether there is a shared component. If so, the two properties can be related through the shared component. Thus, if changes are applied to such shared components, it is possible that changes to properties other than the target property may occur. When implementing design changes, it is desirable to avoid changes to shared components. The relationship between properties and components can be expressed as follows:

$$c_i = f(s_1, s_2, \cdots, s_n), \tag{2}$$

where s_i denotes the n *-th* component that makes up the property c_i, which is composed of design constraint rules f . In the actual product decomposition process, the product components can be a part or a module. The granularity can be determined by the specific research problem.

2.3. Classification of Attributes of Product Properties. Since different properties can be described in different ways, it is necessary to classify the attributes to clearly represent the meaning of various properties. In this paper, the attributes of properties are classified into option-type and parameter-type, which can be expressed as

$$c_i = \left\{ \mathrm{Para}(c_i)_1, \cdots, \mathrm{Para}(c_i)_j, \mathrm{Type}(c_i)_1, \cdots, \mathrm{Type}(c_i)_k \right\}, \tag{3}$$

where $\mathrm{Para}(c_i)_j$ denotes the j *-th* parameter-type attribute of the property c_i and its attribute value can be expressed by a specific parameter value or interval. $\mathrm{Type}(c_i)_k$ denotes the k *-th* option type attribute of the property c_i, and its attribute values are discrete options. For example, the set of moving speed attribute in the moving property of the AGV is {20 cm/s, 30 cm/s, ..., 60 cm/s}. The set of the moving mode attribute should be {wheel type, crawler type}. Only one value can be selected for either parametric or option type attributes.

2.4. Mapping of CR Information to Product Properties. In order to make designers know which components of the product need to be changed, a standard CR template is established to enable the mapping of CRs to product properties.

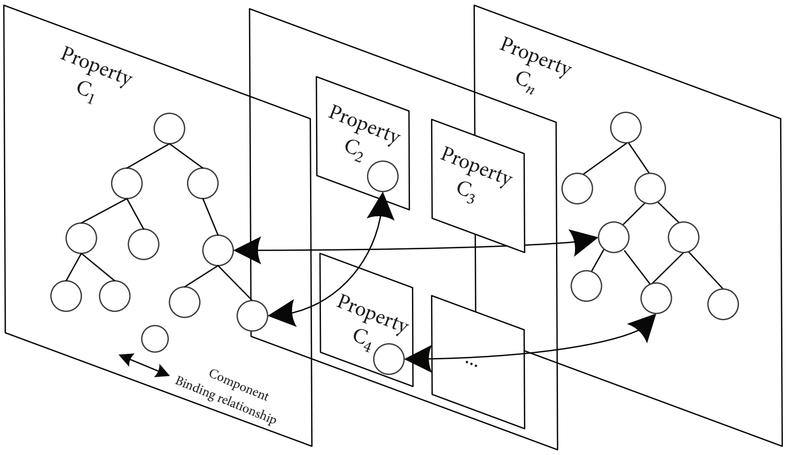

Figure 1: The relationship between properties and components.

The designer should analyse the constituent components of the properties and the constraint rules between the components to determine the final modification based on the template. The CR template can be expressed as

$$\mathrm{CR} = \left\{ d_{ci}, s_{Tj}, s_{Pk} \right\};\, i, j, k \in n, \tag{4}$$

where d_{ci} denotes that the requirement information proposed by the customer is for property c_i, s_{Tj} denotes a description of the j *-th* option-type attribute $\mathrm{Type}(c_i)_j$, and s_{Pk} denotes a description of the k *-th* parameter-type attribute $\mathrm{Para}(c_i)_k$. These values can be determined by the designer based on an understanding of the requirement information or by direct communication with the customer.

For individualized products, their own characteristics make the decomposition of product properties more complicated, which makes it difficult to match CR information with product properties; in this case, semantic similarity can be used to match between requirements and properties. The similarity of two words is determined by the commonality and individuality between words. For example, for any two words ω_1, ω_2, the similarity calculation equation can be expressed as

$$\mathrm{Sim}(\omega_1, \omega_2) = \frac{\alpha}{\alpha + Dis(\omega_1, \omega_2)}, \tag{5}$$

where $\mathrm{Sim}(\omega_1, \omega_2)$ denotes the similarity between ω_1 and ω_2, $\mathrm{Dis}(\omega_1, \omega_2)$ denotes the distance between ω_1 and ω_2, α is an adjustable parameter, and $\mathrm{Dis}(\omega_1, \omega_2)$ can be implemented by Analytic Hierarchy Process [45].

The complete process of mapping CR information to product properties is as follows:

Step 1. Decompose the product at the property level according to the historical change experience and the design knowledge to obtain the set of properties $\{c_1, c_2, \cdots, c_n\}$. Subsequently, the various attributes are classified.

Step 2. In the granularity division of the product components according to the specific research problem, identify the component parts of each property and the corresponding design constraint rule f.

Step 3. Convert the CR information into a standard template $\mathrm{CR}_i = \{d_{ci}, \mathrm{s}_{Tj}, \mathrm{s}_{Pk}\}$.The processed CR information can indicate changes in the product properties. Finally, the relevant components that need to be changed are obtained by combining all the information above.

The model can be expressed as

$$\begin{cases} P = \{c_1, c_2, c_3, \cdots c_n\}, \\ c_i = \left\{ \mathrm{Para}(c_i)_1, \cdots, \mathrm{Para}(c_i)_j, \mathrm{Type}(c_i)_1, \cdots, \mathrm{Type}(c_i)_k \right\}, \\ \mathrm{CR}_i = \left\{ d_{ci}, \mathrm{s}_{Tj}, \mathrm{s}_{Pk} \right\};\, i, j, k \in n, \\ c_i = f(s_1, s_2, \cdots, s_n). \end{cases} \tag{6}$$

3. Product Module Division Based on Community Structure

It is important to build a product part association model. A good descriptive model not only helps customers to understand the structure of the product but also provides a solid basis for quick product configuration. Applying the complex network theory, the parts of the product are treated as network nodes, the correlation between the parts as network edges, and the degree of correlation as the weights of the edges. Thus, an undirected weighted complex network is built, and subsequently, a community detection method based on fuzzy clustering is applied to complete the module division (see Figure 2).

3.1. Correlation Matrix Building. Correlation analysis is to analyse the relationship between parts, which is a prerequisite for module division. Physical correlation, functional correlation, and structural correlation are called correlation items between parts. Their subcorrelation and fuzzy correlation values are shown in Tables 1–3.

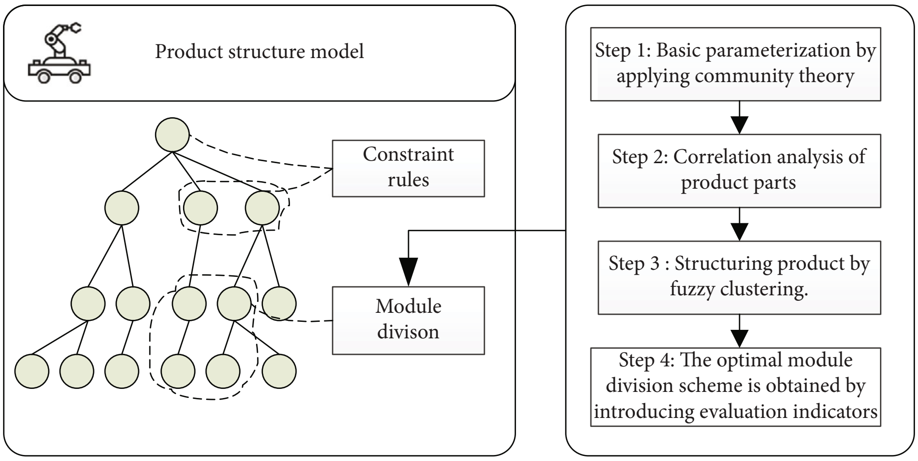
Product structure model
Constraint rules
Module divison
Step 1: Basic parameterization by applying community theory
Step 2: Correlation analysis of product parts
Step 3 : Structuring product by fuzzy clustering.
Step 4: The optimal module division scheme is obtained by introducing evaluation indicators

Figure 2: Individualized product module division method.

Table 1: Physical correlation.

Correlation items	Subcorrelation	Fuzzy correlation values	Description
Physics-related	Energy flow	0.8	Existence of energy transfer relationships, such as power transmission
	Information flow	0.4	Existence of information transfer relationships, such as speed and travel feedback
	Material flow	0.2	Existence of material transfer relationships, such as hydraulic oil
	No relation	0	No physical correlation

Table 2: Structure correlation.

Correlation items	Subcorrelation	Fuzzy correlation values	Description
Structure-related	Permanent bonding	1	Nonremovable bonding such as welding
	Form requirements fit	0.8	Existence of form fit accuracy requirements, such as surface contact requires surface parallelism
	Overfill fit	0.6	The shaft diameter of the mating pair should be larger than the bore diameter
	Transition fit	0.4	Fits that may have gaps or overfills
	Gap fit	0.2	The shaft diameter of the mating pair should be smaller than the bore diameter
	No fit	0	No structural correlation

Table 3: Function correlation.

Correlation items	Subcorrelation	Fuzzy correlation values	Description
Function-related	Strongly related	0.6~0.9	Both are indispensable and work together to achieve the same function
	Weakly related	0.1~0.4	The two are related, the degree of dependence is not particularly large
	No relation	0	No functional correlation

Definition 1. Subcorrelation coefficient c_{ij} between node i and node j is shown as follows:

$$c_{ij} = w_p * p_{ij} + w_f * f_{ij} + w_s * s_{ij}, \tag{7}$$

where p_{ij}, f_{ij}, and s_{ij} are the physical correlation coefficients, functional correlation coefficients, and structural correlation coefficients between node i and node j; e.g., p_{ij} denotes the physical correlation magnitude of node i and node j. The value is the fuzzy correlation value of the subcorrelation in Table 1. w_p, w_f, and w_s are the corresponding weights for physical, functional, and structural correlations; $w_p + w_f + w_s = 1, w_p, w_f, w_s \in [0, 1]$.

According to Tables 1–3 and Definition 1, the physical correlation matrix $P = (p_{ij})_{n\times n}$, the functional correlation matrix $F = (f_{ij})_{n\times n}$, and the structural correlation matrix $S = (s_{ij})_{n\times n}$ are established. The centralized correlation matrix C of products follows that

$$C = w_p * P + w_f * F + w_s * S. \tag{8}$$

Matrix $P, F,$ and S all satisfy the self-reflexivity $p_{ii} = f_{ii} = s_{ii} = 1$ and the symmetry $p_{ij} = p_{ji}, f_{ij} = f_{ji}, s_{ij} = s_{ji}$. Therefore, $c_{ii} = w_p p_{ii} + w_f f_{ii} + w_s s_{ii} = 1, c_{ij} = c_{ji}$; i.e., the centralized correlation matrix C satisfies the self-reflexivity and symmetry.

The subcorrelation coefficients are assigned different weights depending on the object for which the module division is oriented. For example, if the module division is oriented to the customer for configuration, it should be more inclined to the functionality. The allocation of specific weights can be based on expert experience or entropy weight method [46].

3.2. Community Division Based on Fuzzy Clustering. Fuzzy clustering is the aggregation of objects with greater similarity in a data set. Since Ruspinis proposed the concept of fuzzy partitioning in 1969 and made pioneering work in fuzzy clustering analysis, many fuzzy clustering methods have been proposed [47–49]. The transitive closure method of clustering first requires finding fuzzy similarity matrix R. Then, the minimum fuzzy equivalence matrix $t(R)$ is calculated. Finally, the clusters are divided based on $t(R)$.

The centralized correlation matrix C is used as the fuzzy similarity matrix R, $R_{n\times n} = C_{n\times n}$. From Section 3.1, we know that the fuzzy similarity matrix has self-reflexivity and symmetry but does not always have equivalent transferability, so the fuzzy similarity matrix must be transformed. $t(R)$ is a transitive closure matrix transformed from R. According to $t(R)$, we can obtain all possible solutions for the division of the community.

The steps to divide the community are as follows:

Step 1. Find the fuzzy similarity matrix R.$R = (r_{ij})_{n\times n} = C = (c_{ij})_{n\times n}$.

Step 2. Find the transitive closure matrix $t(R)$.

$R \circ R = R^2$, $R^2 \circ R^2 = R^4, \cdots, R^{2^{k-1}} \circ R^{2^{k-1}} = R^{2^k}$. "$\circ$" is the symbol for the multiplication of fuzzy matrix.

When the first occurrence of $R^k \circ R^k = R^k$, $t(R) = R^k$.

Let $U_{n\times t}$ denote the fuzzy matrix with n rows and t columns. For $\forall Q = (q_{ij})_{n\times m}, L = (l_{ij})_{m\times t}, D = Q \circ L \in U_{n\times t}$ and

$$d_{ij} = \bigvee_{l=1}^{m} \left(q_{il} \wedge l_{lj}\right). \tag{9}$$

D is the result of fuzzy multiplication of Q and L. "$\vee$" denotes taking the larger of the two values, and "$\wedge$" denotes taking the smaller one of the two values.

Step 3. Find the λ-cut matrix.

The λ-cut matrix $R_\lambda = (\lambda_{ij})_{n\times n}$ is obtained from $t(R)$,

$$\lambda_{ij} = \begin{cases} 1, & r_{ij} \geq \lambda, \\ 0, & r_{ij} < \lambda, \end{cases} \tag{10}$$

where $i, j = 1, 2, \cdots, n$ and λ is the elemental value of $t(R)$.

Different λ-cut matrices correspond to different community structure division schemes, thus obtaining all possible product module division schemes.

3.3. Preselection of Division Schemes Based on F-Statistic Theory. If all schemes are evaluated in turn, the amount of computation is too large. This will not only waste resources but also reduce efficiency, so a preliminary screening of all division schemes is performed. In this paper, F-statistic theory is used for preselection of module division.

Let the number of communities be r, the number of nodes of the j *-th* community be n_j, the sample of the j *-th* community be $x_1^{(j)}, x_2^{(j)}, \cdots, x_{n_j}^{(j)}$, the cluster centre of the j *-th* community be the vector: $x^{(j)} = (x_1^{(j)}, x_2^{(j)}, \cdots, x_m^{(j)})$, and the mean of the k *-th* variable in the j*-th* community be $x_k^{(j)}$. Therefore,

$$x_k^{(j)} = \frac{1}{n_j} \sum_{i=1}^{n_j} x_{ik}^{(j)} (k = 1, 2, \cdots, m), \quad x^{(j)} = \left(x_1^{(j)}, x_2^{(j)}, \cdots, x_m^{(j)}\right), \tag{11}$$

$$x_k = \frac{1}{n} \sum_{i=1}^{n} x_{ik} \, (k = 1, 2, \cdots, m), \quad x = (x_1, x_2, \cdots, x_m), \tag{12}$$

$$F = \frac{\sum_{j=1}^{r} n_j \left\| x^{(j)} - x \right\|^2 / (r-1)}{\sum_{j=1}^{r} \sum_{i=1}^{n_j} \left\| x_i^{(j)} - x^{(j)} \right\|^2 / (n-r)}. \tag{13}$$

F obeys the F -distribution with degrees of freedom $(r-1, n-r)$. The larger the value of F indicating the lower the coupling between communities, the higher the cohesiveness within the community, and the more reasonable the division of the community. Determine the confidence interval in advance and check the F distribution table. From the theory of multivariate statistical analysis, it is known that if the difference between communities is significant, the division scheme is more desirable. Since each node is classified as a module when λ is maximum and all nodes are classified as a module when λ is minimum, such a division is obviously not practical. The module division scheme corresponding to the maximum and minimum values of λ is eliminated before performing F-statistics.

3.4. Preferential Selection of Division Schemes Based on Modularity

Definition 2. The weight of a node is the sum of the edge weights of the nodes connected to that node.

$$W_i = w_{ij} + w_{ik} + w_{il}, \tag{14}$$

where W_i denotes the weight of node i, nodes $j, k, l, \cdots$ are the nodes connected to node i, and w_{ij} is the edge weight of node i to node j.

The overall quality of the community division needs an evaluation index to measure. Suppose the complex network is divided into m communities, define the $n \times n$ symmetric matrix as e, and use e_{ij} to denote the ratio of the sum of weights between community i and community j to the weights of the whole network. The trace of this matrix is $\mathrm{Tre} = \sum_{i=1}^{m} e_{ii}$, which denotes the ratio of the nodes weight within the i *-th* community to the node weight of the whole network.

The modularity [50] is defined as follows:

$$Q = \mathrm{Tre} - \left\| e^2 \right\|, \tag{15}$$

where $\|e^2\|$ is the modulus of matrix e^2. $Q \in [-1/2, 1]$. The larger the modularity, the more obvious the structure of the community, and the higher the quality of the community division.

4. Configuration of Individualized Products

After the module division is completed, in order to ensure that the design constraint rules between different components are met in new individualized variant product, while companies can fully utilize their previous successful product design experience, this section solves individualized PCP based on DCSP and then queries similar cases through a case-based configuration approach.

4.1. Constraint Relationships between Nodes. Nodes are units in the community network structure, representing parts of the product, and are the smallest units in the product configuration process. In the configuration solving process, the interaction between nodes is realized through various constraint relations, which ensure that the product structure conforms to the design rules. Only when these constraint relations are satisfied, the configuration of the individualized product is reasonable. The constraint relations of nodes can be divided into mandatory constraints, conditional constraints, and exclusive constraints.

(i) Mandatory constraints: IF $N_i \in \Omega$, THEN $N_j \in \Omega$&IF $N_j \in \Omega$, THEN $N_j \in \Omega$.

Mandatory constraints describe a binding relationship that forces association between two nodes, i.e., when node N_i appears in the configuration, node N_j must also appear in the configuration, and vice versa, denoted as $N_i \leftrightarrow N_j$.

(ii) Conditional constraints: $N_i \odot N_j \in \Omega$.

Conditional constraints refer to the relationship between different optional configuration nodes with one-way conditions; i.e., when node N_i is selected, node N_j should be selected at the same time, but the reverse does not necessarily hold, denoted as $N_i \longrightarrow N_j$.

(iii) Exclusive constraints: IF $N_j \in \Omega$, *THEN* $N_j \notin \Omega$.

Exclusive constraints refer to the exclusion relationship between two product configuration nodes; i.e., for the selected set Ω, the configuration nodes N_i and N_j with mutually exclusive relationship cannot exist at the same time, denoted as $x = N_i \longrightarrow y \neq N_j$ (x, y is the corresponding node type in the configuration set Ω).

4.2. Converting PCP to DSCP. CSP is the assignment of respective domain to each variable in a set, so as to find all possible n-tuple elements such that the given constraints can be satisfied [51]. CSP theory has the advantages of being domain independent, descriptive, and simple to use. It has its own mature solution and simplification techniques, which is a suitable method for solving PCP. However, the process of individualized product configuration often contains a lot of dynamic knowledge, such as a variable which can cause the constraints associated with it to appear dynamically due to different assignments by the customer. Therefore, the introduction of DSCP in individualized product configuration makes the whole configuration process dynamic; i.e., the variables involved need not all appear in the solution, and only the activated variables and constraints are involved.

Definition 3. The DCSP is an eight-tuple set:

$$\mathrm{DCSP} = \left(X, D, C, C_T, X_I, X', C', C'_T \right), \tag{16}$$

In the above formula, $X = \{x_1, x_2, \cdots, x_n\}$ is the set of all configuration variables; D denotes the value domain of the variables; C denotes a set of constraints whose elements restrict the occurrence of some variable; C_T denotes the set of constraints that are activated during the solution process; X_I denotes the set of initial variables, which are variables that are necessarily present in the product configuration. During the solution process, the initial variables in X_I are always activated; X' denotes the set of all activated variables in the DCSP solution process. At the initial stage of the solution,$X' = X_I$. As the solution progresses, X' changes dynamically depending on whether a new constraint is activated; C' denotes the set of all activated constraints that have been activated in the configuration process; C'_T denotes the set of constraints that are activated at each step of the product configuration process and are used for the next step.

In the configuration model, each node is mapped as a variable in the DCSP, and the range of values of the node attributes is defined as the domain of the variable. The relationship between nodes is mapped as a constraint between variables, and only nodes that satisfy the design constraint relationships can be combined into products, which is similar to the role of constraints in the DCSP. The CRs can also be regarded as a special constraint. The solution of PCP can be mapped to the solution found by the DCSP (see Table 4).

Solving CR-driven PCP using DCSP can be accomplished using backtracking search algorithms [52]. In addition, a hierarchical constraint approach is proposed to reduce the search space and improve the search efficiency by introducing a hierarchical community structure in complex networks. Before solving the problem, several concepts are defined.

Definition 4. For complex variables, in the product configuration nodes structure, a node with multiple subnodes is called a complex variable. The number of its subnodes can be set according to the product bill of materials.

Definition 5. For hierarchical constraints, the dependencies between different complex variables are called hierarchical constraints. The hierarchical constraint between complex variables x_i and x_j is denoted as $C_{x_i,x_j} = \{(p_s), (q_l)\}$, where $p_s \subset D_{x_i}, q_l \subset D_{xj}$.

As shown in Figure 3, assume that the sets of nodes A and B are compatible and one variable from each of them needs to be taken for combination. There is a hierarchical constraint $C_{a_1,b_1} = \{a_1, b_1\}$ between complex nodes a_1 and b_1. When the set of A is $\{a_{11}, a_{12}, a_{13}, a_{14}\}$, the set of B is restricted as $\{b_{11}, b_{12}\}$. The search space can be reduced by setting hierarchical constraints.

In addition, it is necessary to construct constraint sets and determine the constraint relationships based on the product design knowledge, including establishing hierarchical constraints between complex nodes with multiple subnodes. The solution process is as follows.

Step 1. Assign a value to the initial variables. $X' = X_I = U_0$ (U_0 is the customer's 0 *-th* requirement).

Step 2. Check if new variables and constraints (including hierarchical constraints) are activated. Update X', C', C'_T.

Step 3. Make a new assignment.

Step 4. Repeat Steps 1–3 until all variables have been assigned.

TABLE 4: Conversion of the corresponding elements of PCP and DCSP.

PCP	DCSP
Nodes N_i	Variables X
The sets of values of node attributes	Value domain D
Relationships between nodes N_i	Constraints between variables C
CRs	A type of constraint
Results of the configuration	Constraint optimal solution

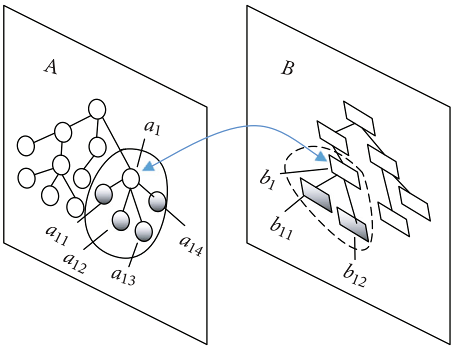

FIGURE 3: Hierarchical constraints between complex variables.

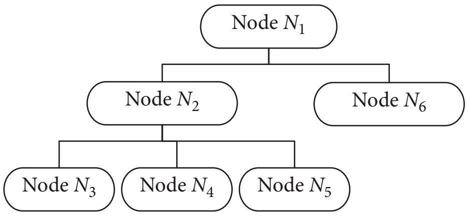

FIGURE 4: Structural importance example.

4.3. Search for Similar Cases. The case-based configuration approach is to store the empirical knowledge in a database and retrieve it when it needs to be used. The advantage is that it enables enterprises to make full use of past successful cases. As long as a similar case is found, it can be partially adjusted by design constraint rules to provide a solution that meets the specific requirements. However, a problem in the case retrieval process is how to discern the similarity between different cases. For this, we use the method of similarity calculation [36] to solve it.

Before calculating the similarity of a new case to an existing case, the structural importance of the j *-th* node should be known first. For convenience, the node at the higher level is set closer to the final product. When the parent node is different, at least one of its subnodes on the child level will change accordingly, and conversely, when the child node changes, its parent node may remain unchanged, so the importance of the node on the parent level is greater than the importance of the node on the child level. The definition of structural importance is as follows.

Definition 6. The sum of the number of nodes N_i itself and its subnodes is the structural importance, denoted as W_i.

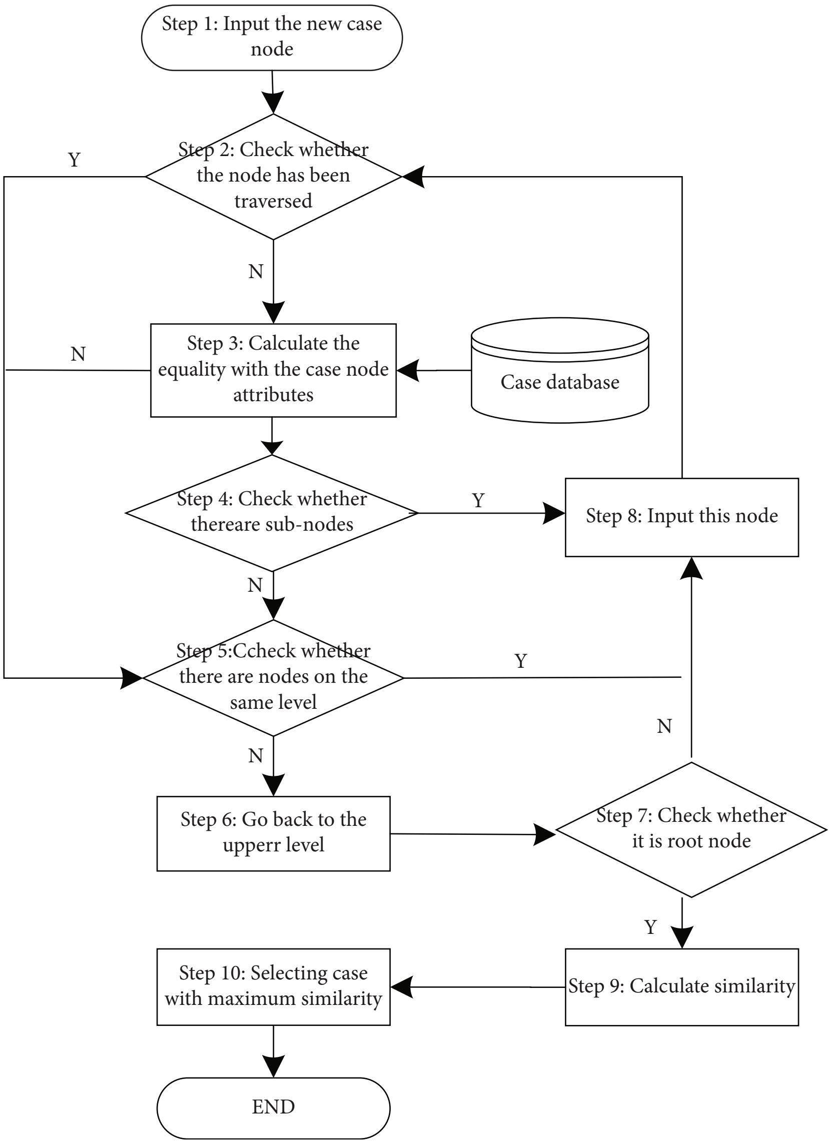
Step 1: Input the new case node
Step 2: Check whether the node has been traversed
Y
N
Step 3: Calculate the equality with the case node attributes
N
Case database
Step 4: Check whether thereare sub-nodes
Y
Step 8: Input this node
N
Step 5:Ccheck whether there are nodes on the same level
Y
N
N
Step 6: Go back to the upperr level
Step 7: Check whether it is root node
Y
Step 9: Calculate similarity
Step 10: Selecting case with maximum similarity
END

FIGURE 5: Similar case search comparison algorithm.

Let m be the number of subnodes of a node. Apparently,

$$W_i = 1 + m. \tag{17}$$

For example, as shown in Figure 4, the structural importance of each node N_i is $W_1 = 6$, $W_2 = 4$, $W_3 = W_4 = W_5 = W_6 = 1$.

The similarity calculation process is as follows:

Step 1. Check whether the node attribute values are the same for different cases.

Let u_j^t denotes the attribute value of the j -th node in case t and $v_{i,j}^D$ denotes the attribute value of the j -th corresponding node of the i -th case in the case database D. When $u_j^t = v_{i,j}^D$, i.e., both nodes are involved in the configuration and have the same attribute value; the output value is 1; otherwise, the output value is 0.

Step 2. Calculate the similarity of each node.

The similarity between the j -th node of a new case t and the j -th node of case i in database D is denoted by $\mathrm{Sim}(u_j^t, v_{i,j}^D)$. The similarity between the nodes of the new case and the existing cases can be calculated by the following equation:

$$\mathrm{Sim}\left(u_j^t, v_{i,j}^D\right) = \frac{\sum_{k=j}^{k=j+n_j-1} S\left(u_k^t, v_{i,k}^D\right)}{n_j}, \tag{18}$$

where n_j is the sum of the number subnodes of the j -th node and itself.

Step 3. Calculate the similarity between new cases and existing cases.

The similarity between the new case and the existing cases can be obtained by accumulating the structural importance multiplied by the similarity and then dividing by the sum of the structural importance of each node.

$$\text{Similary}\left(u_j^t, v_i^D\right) = \frac{\sum_{j=1}^{n}\left[W_j Sim\left(u_j^t, v_{i,j}^D\right)\right]}{\sum_{j=1}^{n} W_j}, \quad (19)$$

where n is the number of case attribute values.

4.4. Case Similarity Comparison. The process of maximum similarity case search algorithm is shown in Figure 5.

Step 1. Input node.

Input the node corresponding to the new cases to be compared with the existing cases.

Step 2. Check whether the node has been traversed.

Yes: jump to Step 5.

No: process Step 3.

Step 3. Check whether the input node and the case node attribute values are equal.

Yes: process Step 4. $S(u_j^t, v_{i,j}^D) = 1$.

No: jump to Step 5. $S(u_j^t, v_{i,j}^D) = 0$.

Step 4. Check whether there are sub-nodes.

Yes: process Step 5.

No: jump to Step 8.

Step 5. Check whether there are nodes at the same level.

Check whether there are no traversed nodes at the same level.

Yes: jump to Step 8.

No: process Step 6.

Step 6. Go back to the upper level.

Step 7. Check if the node is the root node.

Yes: jump to Step 9.

No: process Step 8.

Step 8. Jump to Step 1; input another node for a new comparison.

Step 9. Calculate the similarity.

Step 10. The case with the maximum similarity is selected.

By setting the threshold of similarity, all cases that meet a certain degree of similarity can be presented for selection. In addition, specific design modifications can be made after communication with the customer.

5. Case Study

5.1. An Automated Guided Vehicle Case. AGV is widely used in warehousing, logistics, and other fields with its advantages of high automation, stable system operation, and flexible movements. As a typical individualized product, AGV can be adjusted to its structure according to different application scenarios. In this section, we take an enterprise's stack-type AGV product (see Figure 6) as an example to demonstrate our approach.

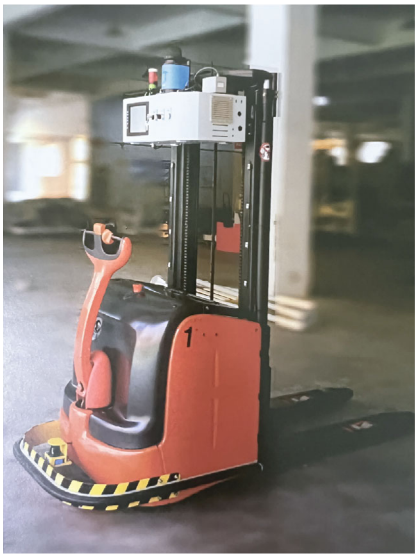

Figure 6: The stack-type AGV.

Through the analysis of the stack-type AGV, it is clear that the main function of this product is to realize the movement of goods in the warehouse. In addition to the movement according to the prescribed path, unlike the translate-type AGV, it is able to move goods up and down. The parts of the AGV are shown in Table 5. Some parts of AGV have been omitted due to article space limitations. They are usually internal connecting parts of the AGV and usually not directly involved in the CRs.

The properties of this stack-type AGV are decomposed into loading property, lifting property, moving property, safety property, and power property. For the completeness of the decomposition, the collection of product parts not covered by the above properties is classified as other property. After the product properties are decomposed, it is necessary to clarify the attributes of each type of properties and further determine their attribute values. Among them, the loading property has only parametric attribute, which is {weight capacity}. Lifting property include option-type attribute and parameter-type attributes, the option-type attribute refers to {lifting mode}, which are {hydraulic type, electric type, pneumatic type}; the parameter-type attributes include {lifting height, lifting speed, lifting weight}. The option type attributes of the moving property include {moving mode, guiding mode}, where the moving mode can be {crawler type, wheel type} and the guide mode can be {magnetic strip type, electromagnetic type, laser type}; the parameter type attribute is {moving speed}. The safety property is expressed as a parametric attribute in terms of {safety factor}. The parametric description of the power property can be determined by {power size}. In summary, the product properties of the stack-type AGV, the properties attributes, and their attribute values are shown in Table 6.

According to the actual development process, the parts corresponding to the product properties of the stack-type AGV are listed to obtain the complete property decomposition (see Figure 7). The drive motor, for example, is a shared part of the lifting property, moving property, and power property.

Through communication with the customer, we understand that the customer is not satisfied with the lifting speed and lifting height of AGV and hopes to improve the efficiency of cargo handling. Collate CRs and express them as

Table 5: AGV parts.

Number	Part name	Number	Part name	Number	Part name
1	Machine housing	6	Drive shaft	11	Operation panel
2	Battery	7	Drive wheel	12	Loading platform
3	Navigation system	8	Reduction gear	13	Lifting actuator
4	Electrical box	9	Brake	14	Lifting bracket
5	Drive motor	10	Directional wheel	15	Limit block

Table 6: Property classification of the stack-type AGV.

Property classification	Parametric-type attributes	Parametric-type attributes value	Option-type attributes	Option-type attributes value
C_1 loading property	Loading capacity	0 kg~300 kg	—	
C_2 lifting property	Lifting speed	150 mm/s ~300 mm/s	Lifting mode	Hydraulic type, electric type, pneumatic type
	Lifting height	2.0 m~5.0 m		
	Lifting weight	1.0 t~4.0 t	—	
C_3 moving property	Moving speed	1 m/s~4 m/s	Moving mode	Wheel type, crawler type
			Guiding mode	Magnetic stripe type, electromagnetic type, laser type
C_4 safety property	Safety factor	0.8~1.0	—	
C_5 power property	Power size	1200 w~1500 w	—	
C_6 other property	—		—	

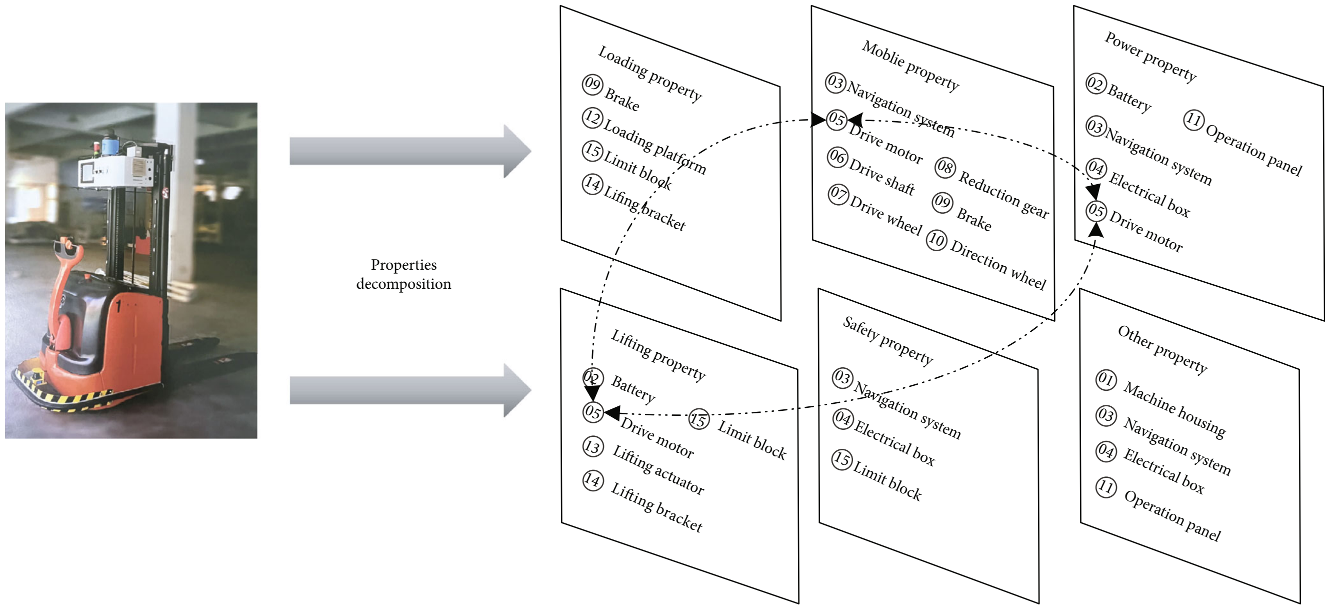

Figure 7: AGV property decomposition.

follows:

$$\mathrm{CRs} = \left\{ \begin{array}{c} \text{Lifting property, Lifting speed, 50\%increase} \\ \text{Lifting property, Lifting height, 30\%increase} \end{array} \right\}. \quad (20)$$

The product part correlation analysis is as follows. Firstly, according to the parts shown in Table 5, each part is viewed as a node. Secondly, correlation analysis is performed based on the fuzzy correlation values in Tables 1–3. Finally, the physical correlation matrix $P = (p_{ij})_{15\times15}$, the functional correlation matrix $F = (f_{ij})_{15\times15}$, and the structural correlation matrix $S = (s_{ij})_{15\times15}$ are established.

The physical correlation matrix is as follows:

$$P=\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 0.8 & 0.4 & 0.8 & 0 & 0 & 0 & 0.4 & 0.4 & 0.4 & 0 & 0.8 & 0 & 0\\
0 & 0.8 & 1 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0.4 & 0 & 0 & 0 & 0 & 0\\
0 & 0.4 & 0.4 & 1 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4 & 0 & 0\\
0 & 0.8 & 0 & 0.4 & 1 & 0.8 & 0.8 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0.8 & 1 & 0.8 & 0.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0.8 & 0.8 & 1 & 0.8 & 0.8 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0.4 & 0.8 & 0.6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0.4 & 0 & 0 & 0 & 0 & 0.8 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0.4 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0.4 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.2 & 0 & 0\\
0 & 0.8 & 0 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0.4 & 0.2 & 1 & 0.2 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}. \tag{21}$$

The functional correlation matrix is as follows:

$$S=\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8 & 0.8 & 0 & 0 & 0\\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 1 & 0.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0.6 & 1 & 0.6 & 0.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0.6 & 1 & 0.4 & 0.8 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0.8 & 0.4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0.8 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\\
0.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\
0.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.8 & 0.8 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8 & 1 & 0.2 & 0.8\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8 & 0.2 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.8 & 0 & 1
\end{pmatrix}. \tag{22}$$

The structural correlation matrix is as follows:

$$F=\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 0.8 & 0.6 & 0.8 & 0 & 0 & 0 & 0.6 & 0 & 0.4 & 0 & 0 & 0 & 0\\
0 & 0.8 & 1 & 0.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0.6 & 0.8 & 1 & 0 & 0 & 0 & 0.2 & 0.4 & 0.4 & 0 & 0 & 0 & 0 & 0\\
0 & 0.8 & 0 & 0 & 1 & 0.8 & 0.6 & 0.2 & 0.2 & 0.2 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0.8 & 1 & 0.8 & 0.4 & 0.4 & 0.2 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0.6 & 0.8 & 1 & 0.4 & 0.4 & 0.2 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0.2 & 0.2 & 0.4 & 0.4 & 1 & 0.6 & 0.2 & 0 & 0 & 0 & 0 & 0\\
0 & 0.6 & 0 & 0.4 & 0.2 & 0.4 & 0.4 & 0.6 & 1 & 0.8 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0.4 & 0.2 & 0.2 & 0.2 & 0.2 & 0.8 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.2 & 0.2 & 0.2\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 1 & 0.8 & 0.4\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.8 & 1 & 0.4\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.4 & 0.4 & 1
\end{pmatrix}. \tag{23}$$

The correlation coefficients are calculated by company design experts based on the entropy weight method: $w_p = 0.2, w_f = 0.6, w_s = 0.2$. The centralised correlation matrix C is obtained from equations (7) and (8).

$$C = 0.2 * P + 0.6 * F + 0.2 * S,$$

$$C=\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.16 & 0.16 & 0 & 0 & 0\\
0 & 1 & 0.64 & 0.44 & 0.64 & 0 & 0 & 0 & 0.44 & 0.08 & 0.32 & 0 & 0.16 & 0 & 0\\
0 & 0.64 & 1 & 0.56 & 0 & 0 & 0 & 0 & 0 & 0.08 & 0 & 0 & 0 & 0 & 0\\
0 & 0.44 & 0.56 & 1 & 0.08 & 0 & 0 & 0.12 & 0.24 & 0.24 & 0 & 0 & 0.08 & 0 & 0\\
0 & 0.64 & 0 & 0.08 & 1 & 0.76 & 0.48 & 0.20 & 0.12 & 0.12 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0.76 & 1 & 0.76 & 0.52 & 0.24 & 0.12 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0.48 & 0.76 & 1 & 0.44 & 0.56 & 0.12 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0.12 & 0.2 & 0.52 & 0.44 & 1 & 0.36 & 0.12 & 0 & 0 & 0 & 0 & 0\\
0 & 0.44 & 0 & 0.24 & 0.12 & 0.24 & 0.56 & 0.36 & 1 & 0.48 & 0 & 0 & 0 & 0 & 0\\
0 & 0.08 & 0.08 & 0.24 & 0.12 & 0.12 & 0.12 & 0.12 & 0.48 & 1 & 0 & 0 & 0 & 0 & 0\\
0.16 & 0.32 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0.08 & 0 & 0\\
0.16 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.32 & 0.28 & 0.12\\
0 & 0.16 & 0 & 0.08 & 0 & 0 & 0 & 0 & 0 & 0 & 0.08 & 0.32 & 1 & 0.56 & 0.4\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.28 & 0.56 & 1 & 0.24\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.12 & 0.40 & 0.24 & 1
\end{pmatrix}. \tag{24}$$

Find the transitive closure matrix:

$$t(R)=\begin{pmatrix}
1 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16\\
0.16 & 1 & 0.64 & 0.56 & 0.64 & 0.64 & 0.64 & 0.52 & 0.56 & 0.48 & 0.32 & 0.16 & 0.16 & 0.16 & 0.16\\
0.16 & 0.64 & 1 & 0.56 & 0.64 & 0.64 & 0.64 & 0.52 & 0.56 & 0.48 & 0.32 & 0.16 & 0.16 & 0.16 & 0.16\\
0.16 & 0.56 & 0.56 & 1 & 0.56 & 0.56 & 0.56 & 0.52 & 0.56 & 0.48 & 0.32 & 0.16 & 0.16 & 0.16 & 0.16\\
0.16 & 0.64 & 0.64 & 0.56 & 1 & 0.76 & 0.76 & 0.52 & 0.56 & 0.48 & 0.32 & 0.16 & 0.16 & 0.16 & 0.16\\
0.16 & 0.64 & 0.64 & 0.56 & 0.76 & 1 & 0.76 & 0.52 & 0.56 & 0.48 & 0.32 & 0.16 & 0.16 & 0.16 & 0.16\\
0.16 & 0.64 & 0.64 & 0.56 & 0.76 & 0.76 & 1 & 0.52 & 0.56 & 0.48 & 0.32 & 0.16 & 0.16 & 0.16 & 0.16\\
0.16 & 0.52 & 0.52 & 0.52 & 0.52 & 0.52 & 0.52 & 1 & 0.52 & 0.48 & 0.32 & 0.16 & 0.16 & 0.16 & 0.16\\
0.16 & 0.56 & 0.56 & 0.56 & 0.56 & 0.56 & 0.56 & 0.52 & 1 & 0.48 & 0.32 & 0.16 & 0.16 & 0.16 & 0.16\\
0.16 & 0.48 & 0.48 & 0.48 & 0.48 & 0.48 & 0.48 & 0.48 & 0.48 & 1 & 0.32 & 0.16 & 0.16 & 0.16 & 0.16\\
0.16 & 0.32 & 0.32 & 0.32 & 0.32 & 0.32 & 0.32 & 0.32 & 0.32 & 0.32 & 1 & 0.16 & 0.16 & 0.16 & 0.16\\
0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 1 & 0.32 & 0.32 & 0.32\\
0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.32 & 1 & 0.56 & 0.40\\
0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.32 & 0.56 & 1 & 0.40\\
0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.16 & 0.32 & 0.40 & 0.40 & 1
\end{pmatrix}. \tag{25}$$

The λ-cut matrices corresponding to $\lambda = 0.16$, $\lambda = 0.32$, $\lambda = 0.40$, $\lambda = 0.48$, $\lambda = 0.52$, $\lambda = 0.56$, $\lambda = 0.64$, $\lambda = 0.76$, and $\lambda = 1$ are calculated sequentially, where $\lambda = 0.40$ corresponds to the λ-cut matrix $R_{0.4}$.

$$R_{0.4}=\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1
\end{pmatrix}. \tag{26}$$

$R_{0.40}$ yields the division scheme as $(1)(2, 3, 4, 5, 6, 7, 8, 9, 10)(11)(12)(13, 14, 15)$.

According to the above calculation results, the community division hierarchy based on fuzzy clustering is shown in Figure 8.

The F-statistic was calculated for all division schemes (see Table 7). The confidence interval α was taken to be 95%. Different F values were derived according to equation (13), and the F-statistic table was checked to obtain $F_{0.05}$. The difference between F and $F_{0.05}$ is F - $F_{0.05}$.

Discard the unreasonable schemes ($\lambda = 0.32$, $\lambda = 0.56$, $\lambda = 0.64$, and $\lambda = 076$), and use the theory of community detection, where the larger the modularity, the better the quality of the community division. From equation (14), $Q_{(\lambda=0.4)} = 0.159453$, $Q_{(\lambda=0.48)} = 0.0860457$, and $Q_{(\lambda=0.52)} = 0.0344529$. $\lambda = 0.4$ corresponds to a larger modularity and is used as the optimal scheme. The network structure of the AGV is shown in Figure 9; the parts represented by the same colour node belong to the same module. The product module associated with the CRs is the lifting module of AGV, including lifting bracket, lifting actuator, and limit block.

For solving the configuration of AGV individualized variant, we constructed a three-level product node structure. The first level is the AGV variant. The second level is the main parameter (para), the interaction device (int-device), and the control device (con-device) associated with the module that needs to be changed. The third level is the specific parts. The total number of variables $j = 10$ which contains individualized variant (AGV), main module parameters ($para$), interaction

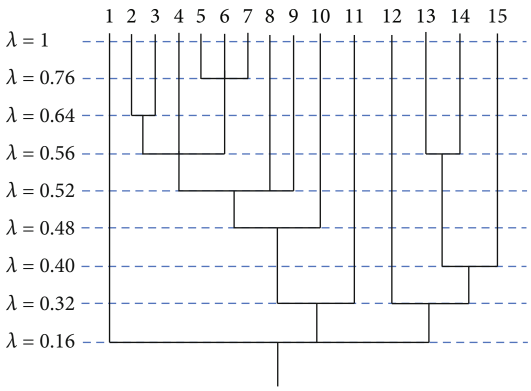

Figure 8: Community division hierarchy.

Table 7: F -statistic value.

λ	Number of modules	F	$F_{0.05}$	F-$F_{0.05}$
0.32	3	3.3130	3.8853	-0.5723
0.40	5	4.3019	3.4780	0.8239
0.48	6	5.3771	3.4817	1.8954
0.52	7	4.4667	3.5806	0.8861
0.56	9	3.5492	4.1468	-0.5976
0.64	12	2.0513	8.7633	-6.7120
0.76	13	0.5217	244.6898	-244.1681

devices (*int-device*), control devices (*con-device*), lifting actuator (*actuator*), lifting bracket (*bracket*), limit block (*block*), operation panel (*panel*), battery (*battery*), and electrical box (*ele-box*). Since the loading platform of this type AGV is used for carrying people and is a universal part, it is ignored in the configuration nodes structure. The hierarchy of configuration nodes structure is shown in Figure 10.

The value range of each variable is as follows:

$$\begin{aligned}
D_{\text{AGV}} &= \{\text{para, int-device, con-device}\},\\
D_{\text{para}} &= \{\text{actuator, bracket, block}\},\\
D_{\text{con-device}} &= \{\text{ele-box, battery}\},\\
D_{\text{int-device}} &= \{\text{panel}\},\\
D_{\text{actuator}} &= \{\text{fast, normal, slow}\},\\
D_{\text{bracket}} &= \{2.0\,\text{m}, 3.0\,\text{m}, 3.5\,\text{m}, 4.0\,\text{m}, 5.0\,\text{m}\},\\
D_{\text{block}} &= \{k_1, k_2, k_3\},\\
D_{\text{ele-box}} &= \{\text{ele-box}_1, \text{ele-box}_2\},\\
D_{\text{ele-box1}} &= \{e_1, e_2, e_4\},\\
D_{\text{ele-box2}} &= \{e_3, e_5, e_6\},\\
D_{\text{battery}} &= \{\text{battery}_1, \text{battery}_2\},\\
D_{\text{battery1}} &= \{b_1, b_3, b_5\},\\
D_{\text{battery2}} &= \{b_2, b_4, b_6, b_7\},\\
D_{\text{panel}} &= \{o_1, o_2, o_3\}.
\end{aligned} \tag{27}$$

There are configuration constraints between the variables as follows:

(1) Conditional constraints:

C_1: actuator→activation variable: block.
C_2: bracket→activation variable: battery.
C_3: battery→activation variable: ele-box.

(2) Mandatory constraints:

C_4 : actuator = “fast” ↔ bracket = “4.0 m” or“ 5.0 m”.
C_5 : bracket = “ 5.0 m” ↔ panel = “ o$_3$”.

(3) Exclusive constraints:

C_6 : actuator = “ fast” ⟶ battery ≠ “ b_1”.
C_7 : actuator = “ fast”&bracket = “4.0 m” ⟶ block ≠ “ k_1 ”.

(4) Hierarchical constraints:

$C_8 : D_{\text{battery,ele-box}} = \{(b_1, b_3, b_5)\ , (e_1, e_2\ , e_4)\}$.

The following are the configuration steps:

Step 1. The system sets the mandatory initial variables according to the product.

($X' = X_I = \{$actuator, bracket, panel$\}$, $C' = \varnothing$, $C'_T = \varnothing$)

Step 2. Select variables according to CR actuator = “fast”: $C'_T = \{C_1, C_2, C_4, C_6\}$.

(-
X' = {actuator = “fast”, bracket = “4.0 m”or”5.0 m”, block, battery, panel}, $C' = \{C_1, C_2, C_4, C_6\}$, $C'_T = \{C_1, C_2, C_4, C_6\}$)

Step 3. Select bracket = “4.0 m”: $C'_T = \{C_7\}$.

(-
X' = {actuator = “fast”, bracket = “4.0 m”, block ≠ ″k_1”, battery ≠ ″b_1”, panel}, $C' = \{C_1, C_2, C_4, C_6, C_7\}$, $C'_T = \{C_7\}$)

Step 4. Select block = “k_2”: $C'_T = \varnothing$.

Step 5. Select battery = “b_3”: $C'_T = \{C_3, C_8\}$.

The activation of the hierarchical constraint C_8 reduces the value space of the variable ele-box.

(-
X' = {actuator = “fast”, bracket = “4.0 m”, block = “k_2”, battery = “b_3”, panel, ele-box}, $C' = \{C_1, C_2, C_4, C_6, C_7, C_3, C_8\}$, $C'_T = \{C_3, C_8\}$)

Step 6. Select panel = “o_1”: $C'_T = \varnothing$.

Step 7. Select ele – box = “e_4”: $C'_T = \varnothing$.

(-
X' = {actuator = “fast”, bracket = “4.0 m”, block = “k_2”, battery = “b_3”, panel = “o_1”, ele-box = “e_4”}, $C' = \{C_1, C_2, C_4, C_6, C_7, C_3, C_8\}$, $C'_T = \varnothing$)

From the above steps, CRs can be mapped to configuration information:

$u^t_j = \{u^t_1\ , u^t_2, \cdots, u^t_{10}\}$ = {AGV, para, actuator = “fast”, bracket = “4.0 m” , block = “k_2”, int-device, panel = “ o_1”, con-device, battery = “b_5 ”, ele-box = “e_4 ”}.

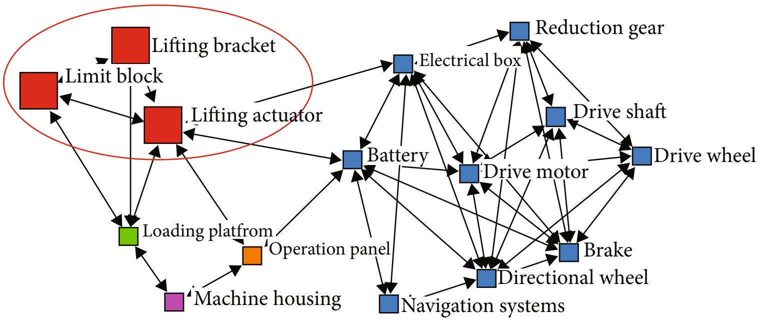

FIGURE 9: The network structure of the AGV.

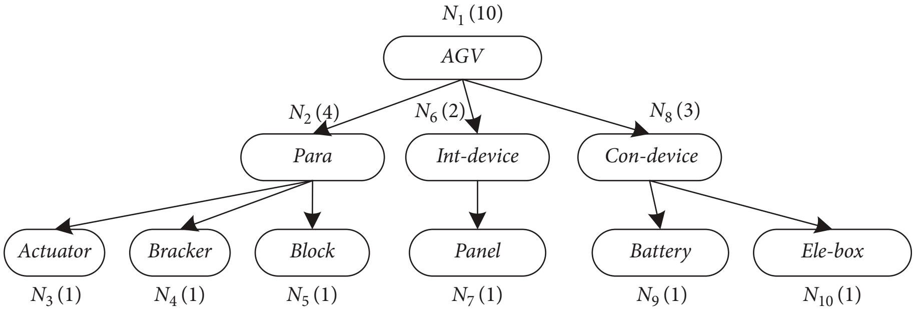

FIGURE 10: Hierarchy of AGV configuration node structure.

The structural importance of each node is calculated from the variable hierarchy and equation (18) as $W_1 = 10$, $W_2 = 4$, $W_3 = W_4 = W_5 = 1$, $W_6 = 2$, $W_7 = 1$, $W_8 = 3$, $W_9 = W_{10} = 1$.

After searching, the similarity of the cases in the case database that best fit the CRs was 76%.

The calculation process is as follows:

Let $i = 1$; then $v_{i,j}^D = \{v_{1,1}^D, v_{1,2}^D, \cdots, v_{1,10}^D\} = \{$transporter, para, actuator = “fast”, bracket = “5.0 m”, block = ″k_3”, int-device, panel = “o_1”, battery = “b_5”, ele-box = “e_4”$\}$.

The node similarity is obtained from equation (17) as follows:

$$\mathrm{Sim}\left(u_j^t, v_{1,j}^D\right) = \frac{\sum_{k=j}^{k=j+n_j-1} S\left(u_k^t, v_{i,k}^D\right)}{n_j} \mid j = 1, 2, \cdots 10 = \left\{\frac{4}{5}, \frac{3}{4}, 1, 1, 0, 1, 1\frac{2}{3}, 0, 1\right\}, \tag{28}$$

when $j = 2$.

$$\mathrm{Sim}\left(u_2^t, v_{1,2}^D\right) = \frac{\sum_{k=2}^{k=2+n_2-1} S\left(u_k^t, v_{1,k}^D\right)}{n_2} = \frac{\sum_{k=2}^{k=5} S\left(u_k^t, v_{1,k}^D\right)}{6} = \frac{1+1+1+0}{4} = \frac{3}{4}. \tag{29}$$

From equation (19), the case similarity is

$$\text{Similary}\left(u_j^t, v_1^D\right) = \frac{\sum_{j=1}^{n} \left[W_j Sim\left(u_j^t, v_{1,j}^D\right)\right]}{\sum_{j=1}^{n} W_j} = \frac{10 \times 4/5 + 4 \times 3/4 + \cdots + 1 \times 0 + 1 \times 1}{10 + 4 + 1 + \cdots + 1 + 1} = 76\%. \tag{30}$$

6. Results and Discussion

As the CRs are only associated with the lifting module of the AGV, the main parameters in the retrieval process are the lifting bracket, lifting actuator, and limit block. After retrieval, the similarity of the configuration solutions that meet the design rules is 76%, which is above the 75% similarity threshold set by the company. The designer presented it to the customer. After communication, the customer agreed to use the solution and expressed satisfaction. This product is the large forward stack-type AGV with a self-weight of 3,400 kg, a rated lifting weight of 3,000 kg, a maximum lifting height of 5.0 m, and a maximum lifting speed of 300 mm/s. The approach is consistent with engineering practice.

In fact, we used this method to create a quick configuration tool using Unity (see Figure 11). The customer can quickly get a product configuration solution by selecting the parameter options. When the customer has questions, the designer will show the network diagram of the product structure and explain why the corresponding parts should be adjusted, which allows the customer to understand the

Figure 11: Quick configuration tool application.

configuration solution visually. The tool can even be used by the customer himself, without the presence of a designer.

However, the method also has some limitations. Due to the module division and product configuration process with some human selected parameters, such as the identification of correlation matrices and design constraint rules, experienced design experts are required to preset them.

7. Conclusion

In the process of individualized product design with customer involvement, uncertain or even unreasonable CRs can lead to the redesign of a large number of product components, which greatly affects the design cycle. To cope with this problem, this paper presents an in-depth study of customer requirements-driven individualized product structure modelling. A community structure-based module division and configuration method is proposed to achieve a fast conversion from CRs to product configuration solutions. Compared with traditional methods, it can describe the product more intuitively due to the use of community structure, allowing customers without specialized knowledge to understand the composition of the product and thus avoiding some unreasonable CRs. This will not be possible if a large Design Structure Matrix or a complex mathematical model is used. In addition, the method has a higher efficiency and can greatly reduce the design cycle, allowing customers to get a satisfactory solution without a long wait.

The paper is concluded as follows:

(1) The concept of properties is used to decompose the product layer by layer, and the attributes of the product properties are divided into two categories: option type and parameter type. By processing the CRs, the information about the product components that need to be changed is obtained

(2) An intuitive network model of individualized products considering physical, structural, and functional correlation is constructed. A community detection method based on fuzzy clustering was applied to achieve modularization. All division schemes were evaluated using the F-statistic and the modularity of the communities, thus improving the accuracy of the module division

(3) PCP is converted to DCSP to address the rationality of the product structure, and a hierarchical constraint is proposed to reduce the search space. The final solution is obtained through a case-based product configuration approach

To improve the robustness of the overall model, there are still some issues that need to be studied in the future. On the one hand, reduce the parameters that need to be manually selected during module division. On the other hand, a few customers may have special requirements for the product, which requires complete redesign of some parts, so the impact assessment of engineering changes will be considered.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (no. 51775156) and the China Scholarship Council (no. 201706695019).

References

[1] Y. Koren, M. Shpitalni, P. Gu, and S. J. Hu, “Product design for mass-individualization,” *Procedia CIRP*, vol. 36, pp. 64–71, 2015.

[2] X. Gu and Y. Koren, “Mass-individualisation – the twenty first century manufacturing paradigm,” *International Journal of Production Research*, pp. 1–16, 2022.

[3] D. Krause, G. Beckmann, S. Eilmus, N. Gebhardt, H. Jonas, and R. Rettberg, “Integrated development of modular product families: a methods toolkit,” in *Advances in Product Family and Product Platform Design: Methods & Applications*, T. W. Simpson, J. (. R.). Jiao, Z. Siddique, and K. Hölttä-Otto, Eds., pp. 245–269, Springer, New York, NY, 2014.

[4] C. Blees, T. Kipp, G. Beckmann, and D. Krause, “Development of modular product families: integration of design for variety and modularization,” in *DS 61: Proceedings of NordDesign 2010, the 8th International NordDesign Conference*, pp. 159–170, Göteborg, Sweden, 2010.

[5] K. Otto, K. Holtta-Otto, T. W. Simpson, D. Krause, S. Ripperda, and S. K. Moon, “Global views on modular design research: linking alternative methods to support modular product family concept development,” *Journal of Mechanical Design*, vol. 138, no. 7, article 071101, 2016.

[6] T. W. Simpson, A. Bobuk, L. A. Slingerland, S. Brennan, D. Logan, and K. Reichard, “From user requirements to commonality specifications: an integrated approach to product family design,” *Research in Engineering Design*, vol. 23, no. 2, pp. 141–153, 2012.

[7] S. J. Hu, J. Ko, L. Weyand et al., "Assembly system design and operations for product variety," *CIRP Annals*, vol. 60, no. 2, pp. 715–733, 2011.

[8] L. W. C. Chen and L. Wang, "Integrating rough set clustering and grey model to analyse dynamic customer requirements," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 222, no. 2, pp. 319–332, 2008.

[9] A. M. M. S. Ullah and J.'i. Tamaki, "Analysis of Kano-model-based customer needs for product development," *Systems Engineering*, vol. 14, no. 2, pp. 154–172, 2011.

[10] J. Kuhl, A. Ding, N. T. Ngo, A. Braschkat, J. Fiehler, and D. Krause, "Design of personalized devices—the tradeoff between individual value and personalization workload," *Applied Sciences*, vol. 11, no. 1, p. 241, 2021.

[11] C. Aguwa, M. H. Olya, and L. Monplaisir, "Modeling of fuzzy-based voice of customer for business decision analytics," *Knowledge-Based Systems*, vol. 125, pp. 136–145, 2017.

[12] L.-S. Chen, C.-C. Hsu, and P.-C. Chang, "Developing a TRIZ-Kano model for creating attractive quality," in *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing, Vols 1-31*, p. 7892, New York, 2008, June 2022, https://www.webofscience.com/wos/alldb/full-record/WOS:000263466103350.

[13] Q. Yang, Z. Ye, X. Li, Z. Zhang, W. Chang, and S. Ruan, "Product module attribute parameter configuration model considering customer requirements preferences," *Mathematical Problems in Engineering*, vol. 2021, Article ID 6632057, 10 pages, 2021.

[14] H. Raharjo, M. Xie, and A. C. Brombacher, "A systematic methodology to deal with the dynamics of customer needs in quality function deployment," *Expert Systems with Applications*, vol. 38, no. 4, pp. 3653–3662, 2011.

[15] Q. Yang, Z. Li, H. Jiao, Z. Zhang, W. Chang, and D. Wei, "Bayesian network approach to customer requirements to customized product model," *Discrete Dynamics in Nature and Society*, vol. 2019, Article ID 9687236, 16 pages, 2019.

[16] M. Nagamachi, "Kansei engineering as a powerful consumer-oriented technology for product development," *Applied Ergonomics*, vol. 33, no. 3, pp. 289–294, 2002.

[17] J. Bonvoisin, F. Halstenberg, T. Buchert, and R. Stark, "A systematic literature review on modular product design," *Journal of Engineering Design*, vol. 27, no. 7, pp. 488–514, 2016.

[18] B. Agard and S. Bassetto, "Modular design of product families for quality and cost," *International Journal of Production Research*, vol. 51, no. 6, pp. 1648–1667, 2013.

[19] R. B. Stone, K. L. Wood, and R. H. Crawford, "A heuristic method for identifying modules for product architectures," *Design Studies*, vol. 21, no. 1, pp. 5–31, 2000.

[20] A. Kusiak and C.-C. Huang, "Development of modular products," *IEEE Transactions on Components, Packaging, and Manufacturing Technology: Part A*, vol. 19, no. 4, pp. 523–538, 1996.

[21] R. Marshall and P. G. Leaney, "Holonic product design: a process for modular product realization," *Journal of Engineering Design*, vol. 13, no. 4, pp. 293–303, 2002.

[22] T. Pimmler and S. Eppinger, "Integration analysis of product decomposition," *Design Theory and Methodology – DTM*, vol. 68, 1994.

[23] G. Erixon, *Modular Function Deployment*, Institutionen för produktionssystem, 1998, July 2022, http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-2619.

[24] M. Aldanondo and E. Vareilles, "Configuration for mass customization: how to extend product configuration towards requirements and process configuration," *Journal of Intelligent Manufacturing*, vol. 19, no. 5, pp. 521–535, 2008.

[25] L. L. Zhang, "Product configuration: a review of the state-of-the-art and future research," *International Journal of Production Research*, vol. 52, no. 21, pp. 6381–6398, 2014.

[26] D. Sabin and R. Weigel, "Product configuration frameworks-a survey," *IEEE Intelligent Systems*, vol. 13, no. 4, pp. 42–49, 1998.

[27] J. McDermott, "RI: an Expert in the Computer Systems Domain," in *Proceedings of the 1st Annual National Conference on Artificial Intelligence, Stanford University*, pp. 269–271, CA, USA, 1980.

[28] Y. Huang, W. K. Ng, H. Liu, W.-F. Lu, B. Song, and X. Li, "Semantic modeling and extraction for cross-family product configuration," in *2007 International Conference on Service Systems and Service Management*, pp. 1–6, Chengdu, China, June 2007.

[29] S. Mittal and B. Falkenhainer, "Dynamic Constraint Satisfaction Problems," in *Proceedings of the 8th National Conference on Artificial Intelligence*, pp. 25–32, Boston, Massachusetts, 1990.

[30] D. Guillon, R. Ayachi, E. Vareilles, M. Aldanondo, E. Villeneuve, and C. Merlo, "Product⋎service system configuration: a generic knowledge-based model for commercial offers," *International Journal of Production Research*, vol. 59, no. 4, pp. 1021–1040, 2021.

[31] D. Yang, J. Li, B. Wang, and Y. Jia, "Module-based product configuration decisions considering both economical and carbon emission -related environmental factors," *Sustainability*, vol. 12, no. 3, 2020.

[32] J. Li and D. Yang, "Optimizing product configuration problems with multisourcing supplier selections under both carbon cap and carbon tax regulations," *Complexity*, vol. 2022, Article ID 6298982, 16 pages, 2022.

[33] Y. Jiao, Y. Yang, and H. Zhang, "An integration model for generating and selecting product configuration plans," *Journal of Intelligent Manufacturing*, vol. 30, no. 3, pp. 1291–1302, 2019.

[34] B. Wielinga and G. Schreiber, "Configuration-design problem solving," *IEEE Expert*, vol. 12, no. 2, pp. 49–56, 1997.

[35] J. Reilly, K. McCarthy, L. McGinty, and B. Smyth, "Dynamic critiquing," *Advances in Case-Based Reasoning, Proceedings*, P. Funk and P. A. G. Calero, Eds., , , pp. 763–777, Springer-Verlag Berlin, Berlin, 2004, June 2022, https://www.webofscience.com/wos/alldb/full-record/WOS:000223825700055.

[36] H.-E. Tseng, C.-C. Chang, and S.-H. Chang, "Applying case-based reasoning for product configuration in mass customization environments," *Expert Systems with Applications*, vol. 29, no. 4, pp. 913–925, 2005.

[37] W. Pengjia, G. Yadong, X. Hualong, L. Yongxian, and A. Y. Nee, "Applying CBR to machine tool product configuration design oriented to customer requirements," *Chinese Journal of Mechanical Engineering*, vol. 30, no. 1, pp. 60–76, 2017.

[38] C. Weber, "Modelling products and product development based on characteristics and properties," in *An Anthology of Theories and Models of Design: Philosophy, Approaches and Empirical Explorations*, A. Chakrabarti and L. T. M. Blessing, Eds., pp. 327–352, Springer London, London, 2014.

[39] D. Krause and N. Gebhardt, "Grundlagen und Begriffe," in *Methodische Entwicklung Modularer Produktfamilien: Hohe Produktvielfalt Beherrschbar Entwickeln*, D. Krause and N. Gebhardt, Eds., pp. 67–88, Springer Berlin Heidelberg, Berlin, Heidelberg, 2018.

[40] J. Zhao, H. Zhou, B. Chen, and P. Li, "Research on the structural characteristics of transmission grid based on complex network theory," *Journal of Applied Mathematics*, vol. 2014, Article ID 261798, 12 pages, 2014.

[41] N.-R. Xu, J.-B. Liu, D.-X. Li, and J. Wang, "Research on evolutionary mechanism of agile supply chain network via complex network theory," *Mathematical Problems in Engineering*, vol. 2016, Article ID 4346580, 9 pages, 2016.

[42] L. Zhang, J. Lu, B. Fu, and S. Li, "A review and prospect for the complexity and resilience of urban public transit network based on complex network theory," *Complexity*, vol. 2018, Article ID 2156309, 36 pages, 2018.

[43] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, article 026113, 2004.

[44] B. Fan, "Modeling of product family stricture and module analysis method based on complex network," *Chinese Journal of Mechanical Engineering*, vol. 43, no. 3, p. 187, 2007.

[45] A. Ishizaka and A. Labib, "Review of the main developments in the analytic hierarchy process," *Expert Systems with Applications*, vol. 38, no. 11, pp. 14336–14345, 2011.

[46] Y. Zhu, D. Tian, and F. Yan, "Effectiveness of entropy weight method in decision-making," *Mathematical Problems in Engineering*, vol. 2020, Article ID 3564835, 5 pages, 2020.

[47] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proc. Fifth Berkeley Sympos. Math. Statist. and Probability (Berkeley, Calif., 1965/66), Vol. I: Statistics*, pp. 281–297, 1967, July 2022, https://mathscinet.ams.org/mathscinet-getitem?mr=0214227.

[48] R. Krishnapuram and J. M. Keller, "The possibilistic C-means algorithm: insights and recommendations," *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 3, pp. 385–393, 1996.

[49] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Springer Science & Business Media, 2013.

[50] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, no. 23, pp. 8577–8582, 2006.

[51] S. C. Brailsford, C. N. Potts, and B. M. Smith, "Constraint satisfaction problems: algorithms and applications," *European Journal of Operational Research*, vol. 119, no. 3, pp. 557–581, 1999.

[52] P. Civicioglu, "Backtracking Search Optimization Algorithm for numerical optimization problems," *Applied Mathematics and Computation*, vol. 219, no. 15, pp. 8121–8144, 2013.