Computers & Industrial Engineering 144 (2020) 106442

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

A novel complex manufacturing business process decomposition approach in cloud manufacturing

Zhenjie Zhang[a,b], Yuanming Zhang[c], Jiawei Lu[c], Fei Gao[c], Gang Xiao[b,c,*]

[a] School of Automation (School of Artificial Intelligence), Hangzhou Dianzi University, Hangzhou, Zhejiang, China
[b] College of Mechanical Engineering, Zhejiang University of Technology, Hangzhou, Zhejiang, China
[c] College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, Zhejiang, China

ARTICLE INFO

Keywords:
Cloud manufacturing
Complex manufacturing business process
Global manufacturing business process network
Decomposition
Complex network

ABSTRACT

In cloud manufacturing, manufacturing business processes are virtualized and encapsulated into manufacturing services. Enterprises can customize their manufacturing business applications on cloud manufacturing platform. To facilitate the customization, a global manufacturing business process is generally built and transformed into a feature model. However, since most global manufacturing business processes have a large number of nodes and complex dependency relationships, it is difficult to transform them into feature models. Therefore, we propose a novel complex manufacturing business process decomposition approach in cloud manufacturing. The approach is beneficial to construct a desired feature model. Firstly, combined with the domain engineering theory, the complex manufacturing business process customization approach in cloud manufacturing is elaborated. Secondly, the complex manufacturing business process decomposition approach is proposed. A global manufacturing business process network (GMBPN) is constructed according to the input and output relationships of manufacturing business activities. Based on the GMBPN, a two-phase decomposition algorithm is presented. In the community detection phase, the nodes in the GMBPN with functional similarity are clustered into the same community with an adaptive strategy. In the sub-process reconstruction phase, the corresponding sub-process is reconstructed in each community. We evaluate the decomposition algorithm with an actual elevator design business process. Experimental results show the proposed algorithm is effective to find the optimal decomposition results.

1. Introduction

To quickly respond to customer diverse requirements, shorten product development cycle and reduce research costs, today's manufacturing enterprises are transforming and upgrading from production-oriented type to service-oriented type (Zhong, Xu, Klotz, & Newman, 2017). Cloud manufacturing (Li et al., 2010; Tao, Zhang, Venkatesh, Luo, & Cheng, 2011; Wu, Thames, Rosen, & Schaefer, 2012; Xu, 2012; Zhang et al., 2014), a new service-oriented manufacturing paradigm, is proposed. It aims to satisfy diversified and personalized requirements of users, and to truly allocate manufacturing resources on demand. In cloud manufacturing, cross-organizational manufacturing resources integration and operation can be realized through virtualization and servitization of manufacturing business processes (Yao et al., 2012, 2013). Enterprises can customize personalized manufacturing business processes on the cloud manufacturing platforms, build online manufacturing business applications and conveniently use them (dynamically or statically binding manufacturing services at run-time).

Usually, the customization of manufacturing business processes in a specific domain is relatively easy to achieve because user requirements often have commonalities and do not change greatly in a period of time. Domain engineering provides a possible solution for domain-oriented business process customization (ESPRIT Consortium AMICE, 1993). It can be applied to build a complete requirement model which can express common requirements and personalized requirements. Feature-oriented modeling method is commonly used in constructing the requirement model for business process customization (Asadi, Mohabbati, Gröner, & Gasevic, 2014). When applying the method, a global business process is generally built and transformed into a feature model. Currently, the construction of feature model mainly relies on domain experts' understanding of the global business process. However, since most global manufacturing business processes have a large number of nodes and complex dependency relationships, it is time-consuming to analyze these complex manufacturing business processes and transform

* Corresponding author at: College of Mechanical Engineering, Zhejiang University of Technology, Hangzhou, Zhejiang, China.
E-mail address: xg@zjut.edu.cn (G. Xiao).

https://doi.org/10.1016/j.cie.2020.106442

Available online 29 March 2020

them into feature models. Meanwhile, it is easy to make mistakes in constructing the feature models if the domain experts cannot understand the global manufacturing business process accurately. As a result, the customization of cloud manufacturing business processes will be affected adversely.

Therefore, this paper aims to provide an approach to reduce the difficulties in analysis of complex cloud manufacturing business processes and construct the feature models more easily and more scientifically. Actually, studies have suggested that decomposing a complex business process into a set of sub-processes (i.e., a small number of nodes and relatively simple dependency relationships) can improve the comprehensibility of the process and reduce the difficulties of analysis (Dikici, Turetken, & Demirors, 2018; Milani, Dumas, Matulevičius, Ahmed, & Kasela, 2016). The sub-processes obtained by an excellent decomposition approach usually represent sub-function of the original complex business process. Through the partial analysis of these sub-processes, the function of the entire complex business process can be more clearly understood (Reijers, Mendling, & Dijkman, 2011). In this regard, designing a scientific and effective decomposition approach to divide the global manufacturing business process into a series of sub-processes is the main problem that needs to be solved. Therefore, a novel complex manufacturing business process decomposition approach in cloud manufacturing based on complex network theory is proposed.

The remainder of this paper is structured as follows. Section 2 discusses the related work. Section 3 elaborates the domain-oriented customization of complex manufacturing business processes. Section 4 introduces the complex manufacturing business process decomposition approach. Section 5 gives the evaluation and discussion of the proposed approach. Section 6 concludes the paper.

2. Related work

2.1. Business process decomposition

Manufacturing business processes are important to enterprise. Users could customize and modify various kinds of business processes through cloud manufacturing platforms (Li et al., 2011; Song, Zhang, Li, & Huang, 2013). Based on the customized manufacturing business processes, the collaboration of upstream and downstream enterprises could be realized by dynamically invoking manufacturing resources (Lin et al., 2016). However, current studies mainly paid attention to the macro-level of manufacturing business process customization (e.g., framework (Schulte, Hoenisch, Hochreiner, Dustdar, Klusch, & Schuller, 2014), platform (Li et al., 2011; Song et al., 2013), system (Ljubicic et al., 2017), etc.), while few concerns the micro-level. Especially, the complexity of manufacturing business processes was lack of consideration. It hindered the customization of complex manufacturing business process in cloud manufacturing.

To improve the understandability and reduce the complexity of analysis of complex business process models, the divide-and-conquer approach was used (Milani et al., 2016). Researchers usually decomposed a complex business process into several sub-processes or modules which were much easier to analyze. It has been proved that structural complexity and modularity were important factors that affected the understandability of business processes (Dikici et al., 2018). Therefore, the study of business process decomposition approach has obvious practical significance. It has been applied to process mining, process modeling, process abstraction, process consolidation and other fields.

The decomposition approach was applied to reduce the complexity of design process in early years (Kusiak & Wang, 1993). It is beneficial to enhance the concurrency and reduce the design cycle. Vanhatalo, Völzer, and Koehler (2009) proposed a business process decomposition approach called Refined Structure Process Tree (RPST), which could be used to decompose the workflow diagram into hierarchical structure and obtain single-entry and single-exit workflow segments with appropriate granularity. Van der Aalst (2012) defined the vertical decomposition and horizontal decomposition. In the vertical decomposition, complete cases were assigned to a group and end-to-end process models were discovered or checked. In the horizontal decomposition, activities were clustered into different groups. After that, a generic approach to decompose the process mining problem (both conformance checking and process discovery) was proposed. For conformance checking, a maximal decomposition mechanism was used to split the overall process model and log into fragments and sublogs that are as small as possible. For process discovery, the researcher also decomposed the overall event log into smaller sublogs and discovered a model fragment for each sublog (Van der Aalst, 2013). Clempner (2010) put forward a hierarchical decision process Petri Net model to solve the problem of complex decision process modeling. By modular modeling, decision process Petri Net was decomposed into multiple simple subnets which could be used to greatly reduce the complexity in the construction of complex systems. Smirnov, Dijkman, Mendling, and Weske (2010) presented a semi-automated business activity clustering method to assist engineers to abstract business process. The semantic relationships of business activities were built through domain ontology. Then the business process was transformed into an abstract model consisting of coarse-grained modules. Although this method considered the semantic information of business activities, it ignored the structural information of the business process. Wiśniewski et al. (2017) modeled the BPMN diagram as a directed graph, and then decomposed the directed graph into different modules by K-gram algorithm to obtain reusable sub-diagrams. The disadvantage of this method was that only structural characteristics of the business process were considered while the semantic information was ignored. Reijers et al. (2011) abstracted the business process into a labeled graph, and then applied the graph clustering method to aggregate business activities. In their research, four optimization metrics (i.e., block-structuredness, connectedness, label similarity, combined connectedness and label similarity) were proposed to evaluate the graph clustering method. The experimental results suggested that the connectedness was helpful to achieve better performance.

In short, existing business process decomposition approaches are limited in solving practical problems of the complex manufacturing business process decomposition and cannot be applied directly in the customization of complex manufacturing business processes. Firstly, the optimal number of sub-processes cannot be obtained automatically. Usually, it needs to be given in advance (Reijers et al., 2011). Secondly, because different researchers usually adopt different decomposition methods and different evaluation metrics, there are no uniform standards for evaluating the quality of process decomposition approach (Johannsen & Leist, 2012; Milani et al., 2016). Therefore, a more effective decomposition approach needs to be established to facilitate the customization of complex manufacturing business processes in cloud manufacturing.

2.2. Application of complex network in cloud manufacturing

The complex network theory could be applied in the modeling and analysis of complex systems. It had attracted much attention in cloud manufacturing (Li, Tao, Cheng, Zhang, & Ne, 2016). For example, Pan and Zheng (2012) built a two-layer trust network model in cloud manufacturing. The upper layer was a Bayesian network and the lower layer was a small-world network composed of virtual enterprise relationships. Cheng, Tao, Zhao, and Zhang (2017) constructed a supply-demand matching hyper-network (SDM) model. The upper layer was the manufacturing task network and the lower layer was the manufacturing service network. Based on the SDM model, manufacturing service selection or manufacturing resources matching could be researched from a new perspective (Tao et al., 2017; Cheng, Bi, Tao, & Ji, 2018, Cheng, Tao, Xu, & Zhao, 2018). Li, Zhu, Wei, Rodrigues, and Wang (2016) constructed a generalized social collaboration network

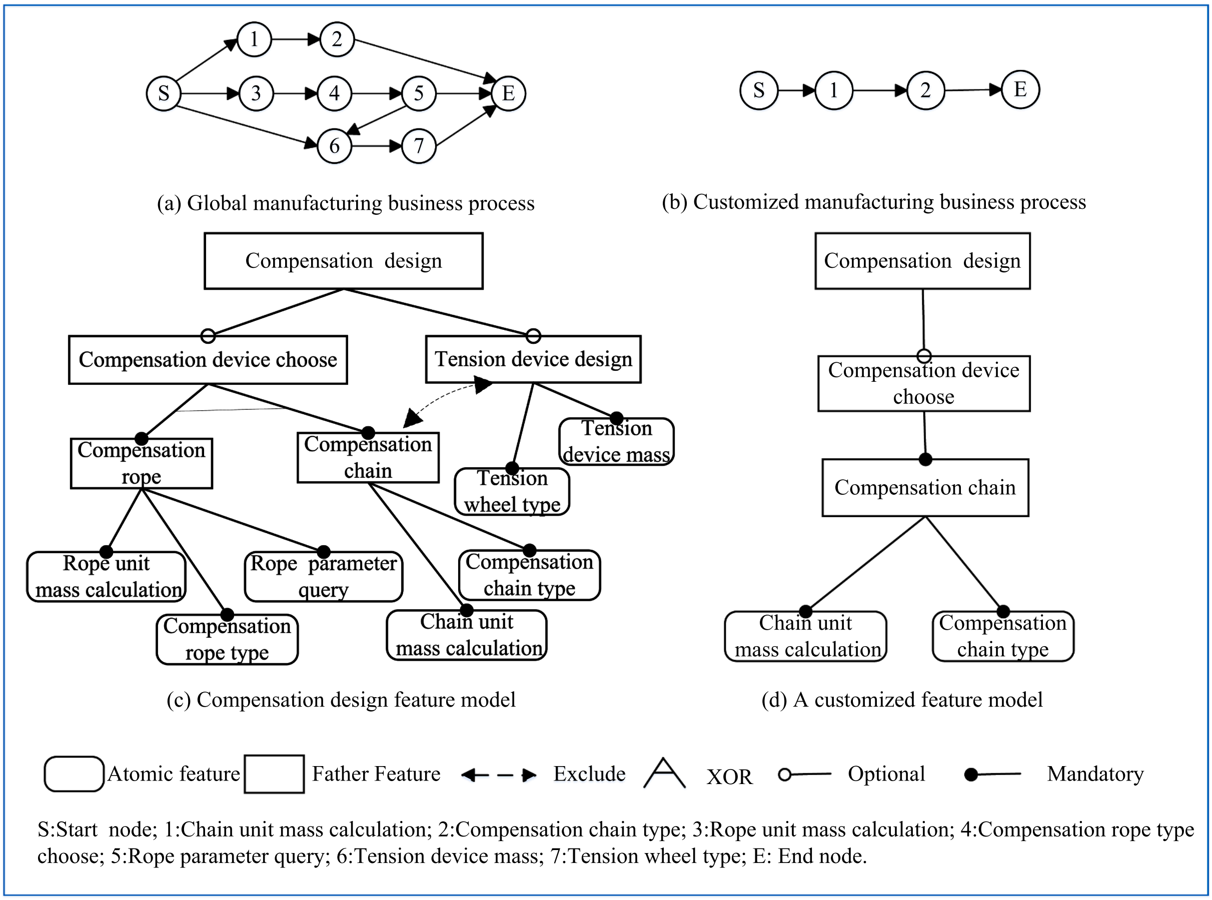


Fig. 1. An example of manufacturing business process customization.

called facilities collaboration network (FCN) to express the collaboration relations in the regional enterprise cluster. Then the dynamic growth process of the FCN was explored when different strategies of facilities selection were used. Ren, Ren, and Jain (2018) studied the social collaboration feature of manufacturing services and established the service social network (SSN) which was used in manufacturing service composition. Geng et al. (2018) applied the complex network theory to analyze the diffusion process and mechanism of cloud manufacturing platform. The simulation results were helpful to the platform operators to make better strategies. Based on the scale-free network model, Liu, Zhang, Tao, and Wang (2013) studied the evolution mechanism of cloud manufacturing systems from the perspective of game theory. Besides, the community detection theory in complex network was also popular in many studies. It was used to explore information exchange (Wu, Schaefer, & Rosen, 2013), cluster the manufacturing services (Wu, He, & Han, 2014) and analyze enterprise relationships (Ding, Jiang, Leng, & Cao, 2016).

The rise of complex network theory provides a new perspective for our research. Modeling and analyzing manufacturing business processes through complex network can provide an automatic tool for better understanding of complex manufacturing business processes. Therefore, the complex network theory is used to solve the problem in the complex manufacturing business process decomposition in this paper.

3. Domain-oriented customization of complex manufacturing business processes

The requirement model is crucial to the customization of manufacturing business processes. For a general requirement model, the requirements are wide-ranging. It makes the modeling process much difficult. Meanwhile, it is adverse to the management of the requirements. Comparatively, domain-oriented customization of manufacturing business processes is relatively easy to realize. The reason is that user requirements in a domain usually have commonalities which can be reused. The concept of the domain refers to a functional area covered by a set of application systems with identical or similar requirements (ESPRIT Consortium AMICE, 1993). For example, elevator design can be seen as a domain. Compared with the general requirement model, the domain requirement model is much easier to establish. Therefore, combined with the authors' previous research (Zhang et al., 2018), the domain-oriented customization of complex manufacturing business processes in cloud manufacturing is presented. To facilitate the elaboration, some related concepts are given as follows.

Definition 1. Global manufacturing business process. It refers to the domain-oriented customization template for a complex manufacturing business process that covers common processes and personalized processes.

Definition 2. Feature model. It refers to the domain requirement model constructed by domain experts based on the global manufacturing business process. The feature model enables complete modeling of manufacturing business process requirements, including common requirements and personalized requirements. The main concepts of feature model are elaborated as follows (Batory, 2005).

Root feature. For each feature model, there is only one root feature. In general, the name of the root feature is the same as the name of the current domain.

Atomic feature. It represents the feature that is abstracted from the manufacturing activity with smallest granularity.

Relationship of features. Relationship of features can be summarized into two categories: refined relationship and constraint relationship. The refined relationship is used to represent a binary relationship between father feature and its sub-features. The refined relationship can be divided into Mandatory feature and Optional feature. Constraint relationship describes the binding relationship between the binding states of a feature. Constraint relationship includes the Require relationship and Exclude relationship between two features, as well as the XOR relationship and OR relationship between father feature and its sub-features.

Based on the global manufacturing business process and the corresponding feature model, manufacturing business process customization can be realized.

Fig. 1 shows a simple example of manufacturing business process customization about the compensation design of elevator.

Fig. 1(a) represents the global manufacturing business process of the elevator compensation design, in which S and E respectively represent the starting node and the terminating node, the other nodes respectively represent a manufacturing business activity, and the arrows represent the dependency relationships of the manufacturing business activities. Fig. 1(c) represents the feature model obtained from the global manufacturing business process in Fig. 1(a). In the feature model, the compensation design represents the root feature and the rope unit mass calculation represents one of the atomic features. There are four constraint relationships in the feature model, for example, compensation design and tension device design is Optional relationship; tension device design and compensation chain is Exclude relationship; tension device design and tension device mass is mandatory relationship; compensation rope and compensation chain is XOR relationship. Fig. 1(d) represents a customized feature model tailored from the compensation design feature model in Fig. 1(c). Fig. 1(b) represents the customized manufacturing business process according to the customized feature model in Fig. 1(d).

Definition 3. Complex manufacturing business process. It refers to the manufacturing business process model with a large number of business activity nodes and complex business activity dependency relationships.

According to the above elaboration, the complex business process customization approach in cloud manufacturing is shown in Fig. 2.

It can be described with three phases: global manufacturing business process modeling, feature model construction and personalized manufacturing business process customization.

In the first phase, manufacturing process data is acquired from existing manufacturing business systems, manufacturing business process documents or standard documents of enterprises. Then the acquired process data is standardized by the domain dictionary which is built by domain experts. Finally, a standard global manufacturing business process is built.

In the second phase, a feature model based on the global manufacturing business process is constructed as a domain requirement model for manufacturing process customization. At the same time, the mapping relationship between the feature model and the global manufacturing business process is established.

In the third phase, a customized feature model is obtained based on the user's customization requirements. Then the customized feature model is transformed into an executable manufacturing business process according to the mapping relationship.

As shown in Fig. 2, the key phase of complex business process customization in cloud manufacturing is to construct the feature model based on the global manufacturing business process. Therefore, the main problems that need to be solved in this paper are concentrated in the second phase. It can be summarized as follows: for a global manufacturing business process, an automatic process decomposition approach is needed to obtain the optimal sub-processes and thus facilitate the construction of feature model.

4. Complex manufacturing business process decomposition approach

To realize the decomposition of a global manufacturing business process, this section studies the complex manufacturing business process decomposition approach on the basis of complex network. First, the complex network is used to construct the global manufacturing business process and the concept of global manufacturing business process network (GMBPN) is proposed. Second, a two-phase decomposition algorithm is designed to decompose the GMBPN.

4.1. Global manufacturing business process network

Definition 4. Global manufacturing business process network. It refers to the domain-oriented manufacturing business process template. The GMBPN is a complex network model with nodes representing manufacturing business activities and edges representing dependency relationships of manufacturing business activities. Its formal description is as follows.

(1) Atomic manufacturing business activity. It refers to the standard manufacturing business activity with minimum granularity. It can be described as a multi-tuple: MBPNode = {ID, Name, Input, Output, Description}, where ID represents a unique identifier for manufacturing business activity, Name represents the name of the manufacturing business activity, Input represents the input information of manufacturing business activity, Output represents the output information of manufacturing business activity, Description represents the textual description of manufacturing business activity which can be used to extract domain labels.
(2) Manufacturing business activity dependency edge. It refers to the logical dependency relationship between two manufacturing business activities. It can be recorded as: MBPEdge = {StartNode, EndNode}, where the output of StartNode is part or whole of the input of EndNode.
(3) Global manufacturing business process network. It consists of a set of atomic manufacturing business activities and manufacturing business activity dependency edges. It can be recorded as: MBPNet = {DomainName, MBPNodes, MBPEdges}, where DomainName represents the name of the domain, MBPNodes represents the set of manufacturing business activity nodes and MBPEdges represents the set of manufacturing business activity dependency edges.

4.2. GMBPN decomposition algorithm

Based on the GMBPN, the decomposition mechanism is shown in Fig. 3. It contains two phases: community detection and sub-process reconstruction.

In the community detection phase, the nodes with functional similarity are clustered into the same community. In the sub-process reconstruction phase, the corresponding sub-process is reconstructed according to the logical dependencies of the manufacturing business activities in each community. According to the GMBPN decomposition mechanism, the corresponding algorithm contains two parts: community detection algorithm and sub-process reconstruction algorithm.

4.2.1. Phase 1: Community detection of the GMBPN

The community detection results are very important because it is the foundation of the sub-process reconstruction. In complex network, community detection aims to cluster the nodes with functional

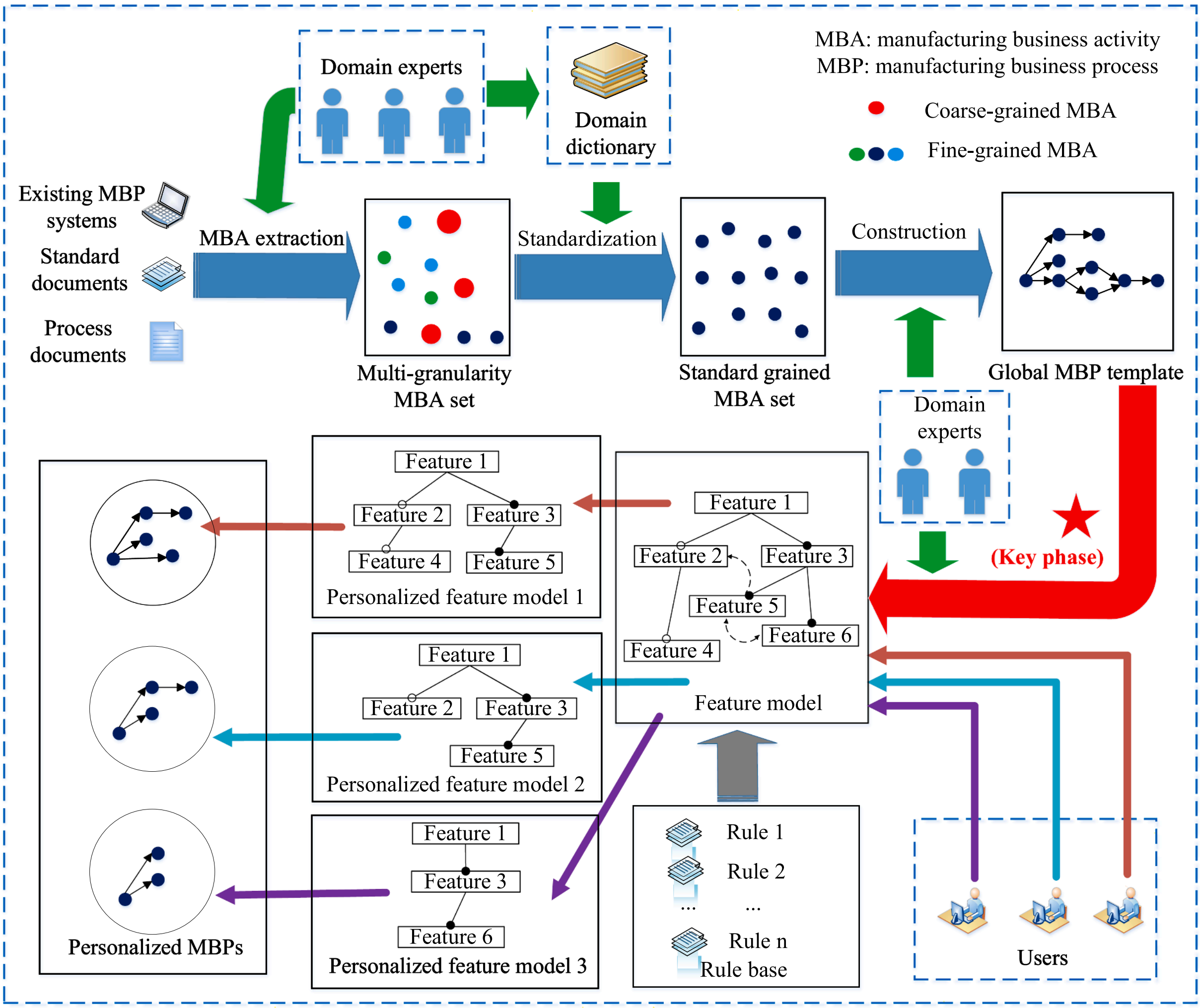


Fig. 2. Domain-oriented cloud manufacturing business process customization.

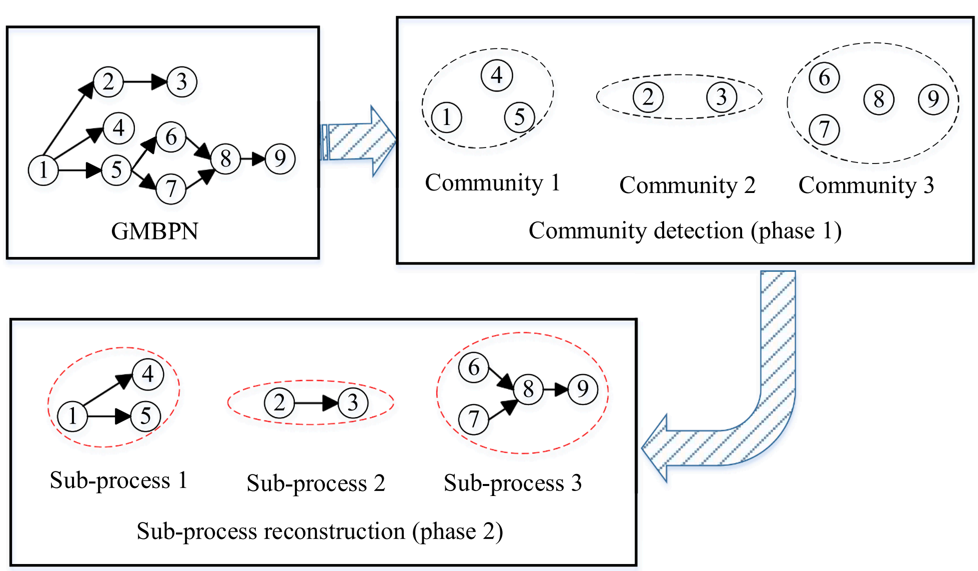


Fig. 3. GMBPN decomposition mechanism.

similarity into the same category. The functional similarity reflects in two aspects: first, the nodes in the same community are closely connected, and the connections of different communities are sparse; second, the similarity of nodes in the same community is high, and the similarity of different communities is low. Based on this criterion, the nodes in GMBPN can be divided into several communities.

Presently, the community detection algorithms in complex network can be mainly categorized into three types: topology-based, attributes-based, combined topology and attributes. However, the topology-based algorithms fail to make full use of the node attribute information and the attribute-based algorithms ignore the importance of structure information of a network. For this reason, both types of algorithms have limitations and often cannot obtain good detection results. Therefore, algorithms which merge network topology information and node attribute information are receiving more and more attention from researchers. Nevertheless, the main problem of these algorithms is that much priori information needs to be provided. For example, the optimal weights of network topology and node attribute are required (Tang & Viennet, 2012), or the number of communities needs to be provided in advance (Cheng, Zhou, & Jeffery, 2011). Actually, the information is difficult, sometimes even impossible, to obtain. Therefore, these algorithms cannot be applied directly in the community detection of the GMBPN.

For the community detection of the GMBPN, the following two problems need to be considered.

Firstly, since the optimal number of communities is difficult to provide in advance, how to automatically determine the optimal number of communities? The optimal number of communities means that the number is appropriate and the theme of each community is clear.

Secondly, how to correctly and reasonably use the network topology information and the node attribute information to calculate the similarity of the business activity nodes, that is, how to determine the weights of the information properly?

Before designing the algorithm, some basic concepts are introduced as follows.

(1) **Modularity**

Modularity is used to measure the quality of community detection (Newman & Girvan, 2004). The goal of community detection is to make the connections within the same community relatively close, and the connections among different communities relatively sparse. The greater the modularity, the better the community detection results. The calculation of the modularity is shown in Eq. (1):

$$Q = \frac{1}{2m}\sum_{i,j}\left(A_{ij} - \frac{K_i K_j}{2m}\right)\delta(C_i, C_j) \tag{1}$$

where Q represents the modularity; A_{ij} represents the weight between nodes i and j; K_i and K_j respectively represent the sum of the weights of node i and node j in the network; C_i and C_j respectively represent the

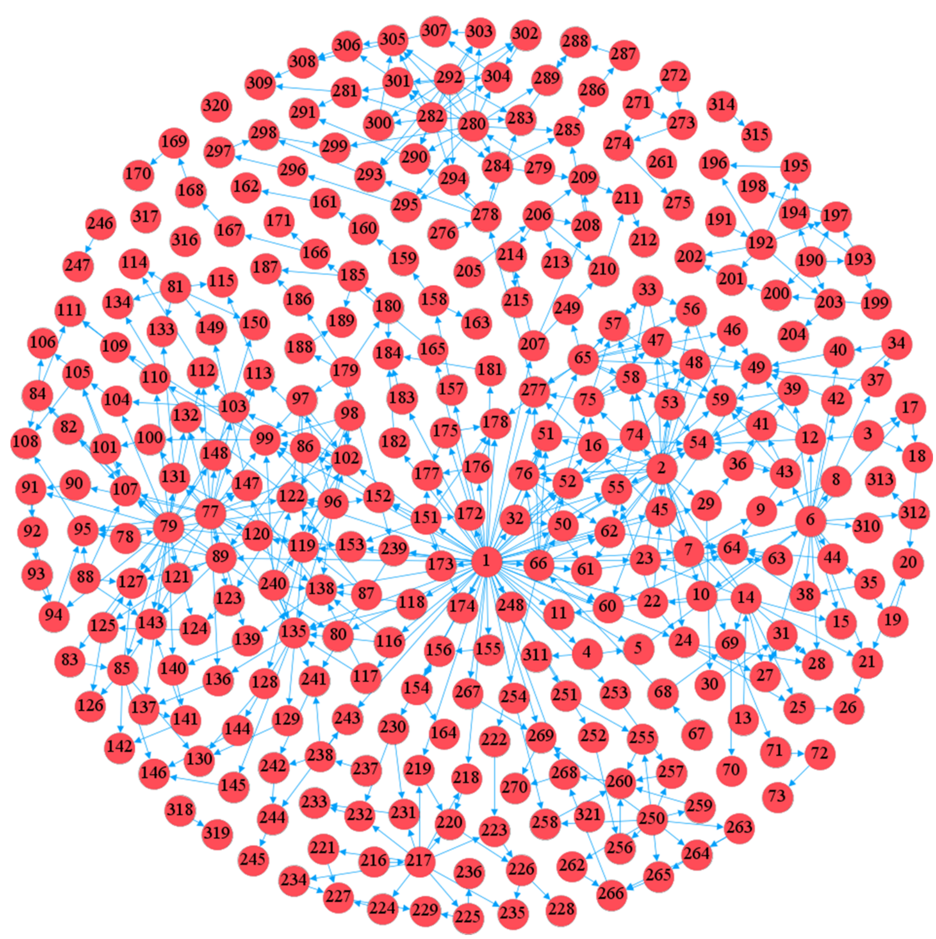


Fig. 4. The GMBPN of elevator design.

Table 1
Characteristics comparison of different algorithms.

Algorithm	Characteristics			
	Network topology information	Node attribute information	Optimal number of communities	Weight information
Louvain	√	×	×	—
FN	√	×	×	—
H-Cluster	×	√	√	—
SAC	√	√	×	√
SA-Cluster	√	√	√	√
HAM	√	√	√	√
CD-GMBPN	√	√	×	×

community which node i and node j belong to, if i and j are in the same community, then $\delta(C_i, C_j) = 1$, otherwise $\delta(C_i, C_j) = 0$; m is the sum of the weights of all the edges in the network.

(2) **Modularity increment**

The modularity increment is a concept proposed in the Louvain algorithm (Blondel, Guillaume, Lambiotte, & Lefebvre, 2008), which is one of the best community detection algorithms. The basic idea of the algorithm is to divide the community by modularity increment ΔQ. When a node is moved to different communities, the changes of modularity increment are different. When the node i is moved to a community C, the resulting modularity increment ΔQ can be calculated by Eq. (2):

$$\Delta Q = \left[\frac{\Sigma_{\rm in} + K_{i,\rm in}}{2m} - \left(\frac{\Sigma_{\rm tot} + K_i}{2m}\right)^2\right] - \left[\frac{\Sigma_{\rm in}}{2m} - \left(\frac{\Sigma_{\rm tot}}{2m}\right)^2 - \left(\frac{K_i}{2m}\right)^2\right] \tag{2}$$

where $\Sigma_{\rm in}$ is the sum of the weights of the edges inside C, $\Sigma_{\rm tot}$ is the sum of the weights of the edges incident to nodes in C, K_i is the sum of the weights of the edges incident to node i, $K_{i,\rm in}$ is the sum of the weights of the edges from i to nodes in C and m is the sum of the weights of all the edges in the network.

Based on the concept of modularity increment, the community detection algorithm of the GMBPN is proposed. The algorithm can optimize the modularity by dynamically adjusting the weights of network topology information and the node attribute information, and thus obtain the optimal detection results. The specific implementation processes are as follows.

Step 1: Extracting network topology information of the GMBPN and constructing the structural similarity matrix A. The element A_{ij} in the matrix A represents the structural similarity of node i and node j. If there is an edge between node i and node j, $A_{ij} = 1$; otherwise, $A_{ij} = 0$.

Step 2: Extracting node attribute information of the GMBPN and constructing the domain similarity matrix B. The element B_{ij} in the matrix B represents the domain similarity of node i and node j. The domain similarity between node i and node j can be calculated by Eq. (3):

$$B_{ij} = \frac{L(i) \cap L(j)}{L(i) \cup L(j)} \tag{3}$$

where $L(i)$ and $L(j)$ represent the attribute label vectors of node i and node j respectively. The attribute label vector consists of domain labels which can be obtained through the domain dictionary.

Step 3: According to the obtained structural similarity matrix A and domain similarity matrix B, to construct the composite modularity as the optimization target. The composite modularity can be calculated by Eq. (4):

$$Q_c = \alpha Q_s + \beta Q_a \tag{4}$$

where Q_s and Q_a respectively represent the modularity generated by the network topology and node attribute, which can be calculated according to Eq. (1); α and β represent the weights of the topology and attributes respectively.

Step 4: Finding out the communities of the GMBPN based on the modularity increment ΔQ_c. It can be calculated by Eq. (5):

$$\Delta Q_c = \alpha \Delta Q_s + \beta \Delta Q_a \tag{5}$$

where ΔQ_s and ΔQ_a can be calculated according to Eq. (2).

Step 5: Obtaining a new network according to the results of **step 4**, and calculating the new Q_c. If Q_c is unchanged, the community detection process is finished; otherwise, replying **step 4**.

In reality, the optimal values of α and β are difficult to determine in advance. Therefore, in our algorithm, an adaptive strategy is introduced to adjust the values automatically. During the process of community detection, it is reasonable assumed that the differences of each community's theme should become larger and larger. To measure the community's theme, an evaluation metric called theme difference coefficient (*TDiff*) is proposed as another optimization target. *TDiff* consists of two secondary metrics: *Glo-TDiff* and *Min-TDiff*. *Glo-TDiff* is a global indicator which can be calculated by Eqs. (6) and (7). *Min-TDiff* is a local indicator which refers to the minimum value of any two communities and can be calculated by Eq. (8). The larger the two metrics, the better the community detection results.

$$Glo - TDiff = \frac{2\sum_{u=1}^{N-1}\sum_{v=u+1}^{N}[TD(u, v)]}{N(N-1)} \tag{6}$$

$$\begin{aligned} TD(u, v) &= 1 - d(a(u), a(v)) \\ a(u)_i &= p(x(t), u)_i \end{aligned} \tag{7}$$

$$Min\text{ - }TDiff = \min\{TD(u, v)\} \tag{8}$$

where N is the total number of communities; u, v is the community index; $TD(u, v)$ is the theme difference coefficient of community u and community v; $a(u)$ and $a(v)$ are attribute probability vectors of community u and community v respectively; $d(a(u), a(v))$ represents the similarity of $a(u)$ and $a(v)$ that calculated by cosine distance; x represents the attribute vector; $p(x(t), u)$ represents the proportion of the t dimension attribute in the community u.

Here we use the *Glo-TDiff* as the optimization target. Firstly, it is assumed that the initial values of α and β are the same. Then we set a suitable weight increment $\Delta\omega$ to adjust the values of α and β, as shown in Eq. (9).

$$\begin{cases} \alpha = \alpha + \Delta\omega & if\ \ Glo - TDiff\,(i) - Glo - TDiff\,(i-1) < 0 \\ \beta = \beta + \Delta\omega & if\ \ Glo - TDiff\,(i) - Glo - TDiff\,(i-1) > 0 \end{cases} \tag{9}$$

If *Glo-TDiff* is in an increasing state, it indicates that the composite modularity (Q_c) is optimized and the community's theme is also getting better and better. Therefore, the attribute information is considered to have a positive effect. So the value of β should be appropriately increased; On the contrary, the attribute information has an obstructive effect, and it is necessary to increase the value of α. Based on the change of *Glo-TDiff*, the value of α and β can be adjusted automatically.

The community detection algorithm of the GMBPN (CD-GMBPN algorithm) is shown as follows.

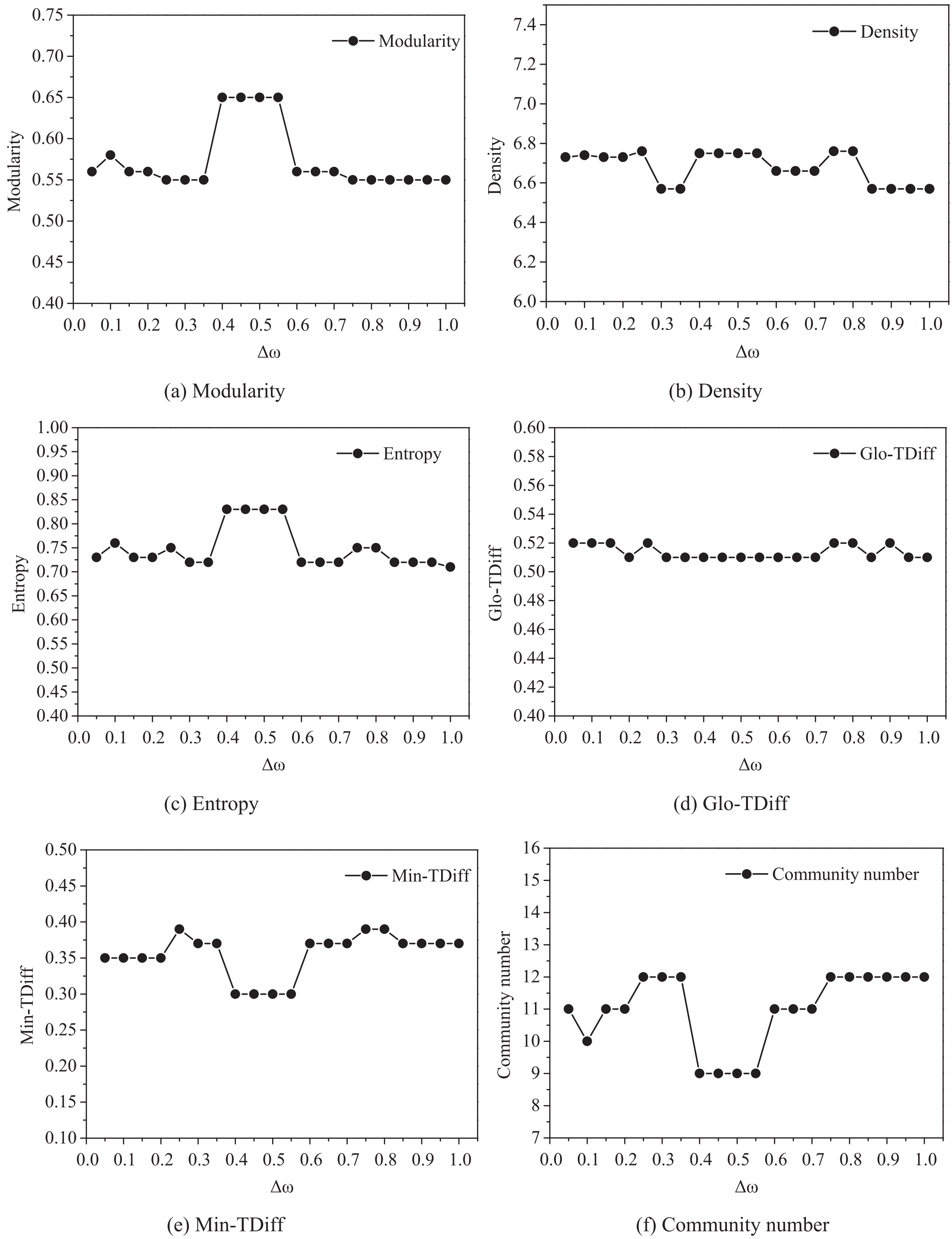


Fig. 5. Sensitivity analysis of initial parameters.

Community detection algorithm of the GMBPN.

```
Input: structure similarity matrix A, domain similarity matrix B, original weight of
    network topology information α, original weight of node attribute information β,
    weight increment parameter Δω
Output: CNode = {C1, C2, …, Cn}
CNode = Initialize(Node);
  WHILE LQc > 0
  WHILE index = true
  index ← false;
  N = getSize(CNode); // obtain the number of communities
  FOR i = 1:N do
    K[i] = getWeight(CNode); // obtain the degree of nodes
  END FOR
  K[i] = descendRank(K[i]);
    CNode = updateCNode(K); // sort by the degree of nodes
      FOR V[u] ∈ CNode do
        FOR V[w] ∈ CNode do
          ΔQc[u][w] = deltaQCalculate(α, β, A, B); // according to Eq.(1)&
(2)&(4)&(5)
        END FOR
        [w, maxΔQc] = Max(ΔQc[u][w]);
        IF maxΔQc > 0 THEN
          index ← true;
          C(V[w]) ← V[u];
          Glo-TDiff = calculateGloTDiff(a(x)); // according to Eqs.(6)&(7)
          α = updateAlpha(α, Δω, Glo-TDiff);
          β = updateBeta(β, Δω, Glo-TDiff); // according to Eq.(9)
        ELSE
          C(V[u]) ← V[u];
          END IF
        END FOR
        CNode = updateCommunity(CNode);
      END WHILE
        A = updateMatrix(A);
        B = updateMatrix(B);
        Qc[t] = calculateModularity(α, β, A, B);
        LQc = Qc[t]- Qc[t-1];
  END WHILE
  Return CNode = {C1, C2, …, Cn}
```

4.2.2. Phase 2: sub-process reconstruction

Based on the result of the community detection, the corresponding sub-process is reconstructed according to the logical dependency relationship of the manufacturing business activities in each community. The sub-process reconstruction algorithm is shown as follows.

Sub-process reconstruction algorithm.

```
Input: business process complex network matrix CM, community detection results
    {C1, C2, …, Cn}
Output: sub-processes matrix{CM1, CM2, …, CMn}
FOR u = 1:n do
  num = getSize(C[u]); //obtain number of nodes in each community
  FOR i = 1:num do
    FOR j = 1:num do
      IF Output(Node[i]) ∩ Input(Node[j]) = true THEN //matching of ou-
tput and input information
        CM(i, j) ← 1;
      ELSE
        CM(i, j) ← 0;
      END IF
    END FOR
  END FOR
END FOR
Return {CM1, CM2, …, CMn}
```

5. Evaluation and discussion

Based on the cloud manufacturing platform for the elevator industry alliance developed by authors' group (Zhang et al., 2018), the effectiveness of the proposed approach is verified. Through the cloud manufacturing platform, different elevator enterprises can customize personalized manufacturing business processes on demand and build online business applications for elevator design.

Table 2
Metrics obtained by different algorithms (the optimal values are bold and underlined).

Algorithm	Community number	*Modularity*	*TDiff*		*Fusion Value*
			Glo-TDiff	*Min-TDiff*	
Louvain	21	**<u>0.72</u>**	0.56	0.19	0.87
FN	21	0.71	**<u>0.57</u>**	0.11	0.79
SAC	7	0.61	0.46	0.22	0.80
CD-GMBPN	9	0.65	0.51	**<u>0.30</u>**	**<u>0.93</u>**

At present, there are 321 standard elevator design business activities. According to the logical dependency relationships between elevator design business activities, the GMBPN of elevator design is constructed (as shown in Fig. 4). In the GMBPN of elevator design, each node represents a standard elevator design activity and each arrow represents the dependency relationship between two activities. Then, the proposed algorithm is applied to decompose the GMBPN of elevator design.

5.1. Algorithm evaluation

In our experiments, six typical algorithms are chosen to compare with the CD-GMBPN algorithm. Table 1 compares the characteristics of different algorithms.

Algorithm 1 is the Louvain algorithm, which is a classical community detection algorithm based on modularity (Blondel et al., 2008). Algorithm 2 is the FN algorithm, which is also a classical community detection algorithm based on modularity (Newman, 2004). In these two algorithms, only the network topology information is considered. Algorithm 3 is the SAC algorithm, in which both node attribute information and network topology information are considered (Tang & Viennet, 2012). Algorithm 4 is the classical hierarchical clustering algorithm (H-Cluster algorithm), in which only the node attribute information is used. Algorithm 5 is the SA-Cluster algorithm, in which both node attribute information and network topology information are considered (Cheng et al., 2011). Algorithm 6 is the HAM algorithm, in which the node attribute information and business process structure information are considered (Reijers et al., 2011).

To evaluate the performance of different algorithms, four metrics (i.e., *Modularity* (Newman, 2004), *Density* (Cheng et al., 2011), *Entropy* (Cheng et al., 2011), *TDiff*) are used.

(1) ***Modularity***

The *Modularity* can be calculated according to Eq. (1). The larger the value of the *Modularity*, the better the community detection algorithm.

(2) ***Density***

The *Density* can be calculated according to Eq. (10).

$$Density(\{C_i\}_{i=1}^{k}) = \sum_{i=1}^{k} \frac{|\{(v_p, v_q)|v_p, v_q \in C_i, (v_p, v_q) \in E\}|}{|E|} \tag{10}$$

where $|E|$ represents the number of edges in the network; C_i represents the community; k represents the number of communities. The larger the value of the *Density*, the better the community detection algorithm.

(3) ***Entropy***

The calculation of the *Entropy* is shown in Eq. (11).

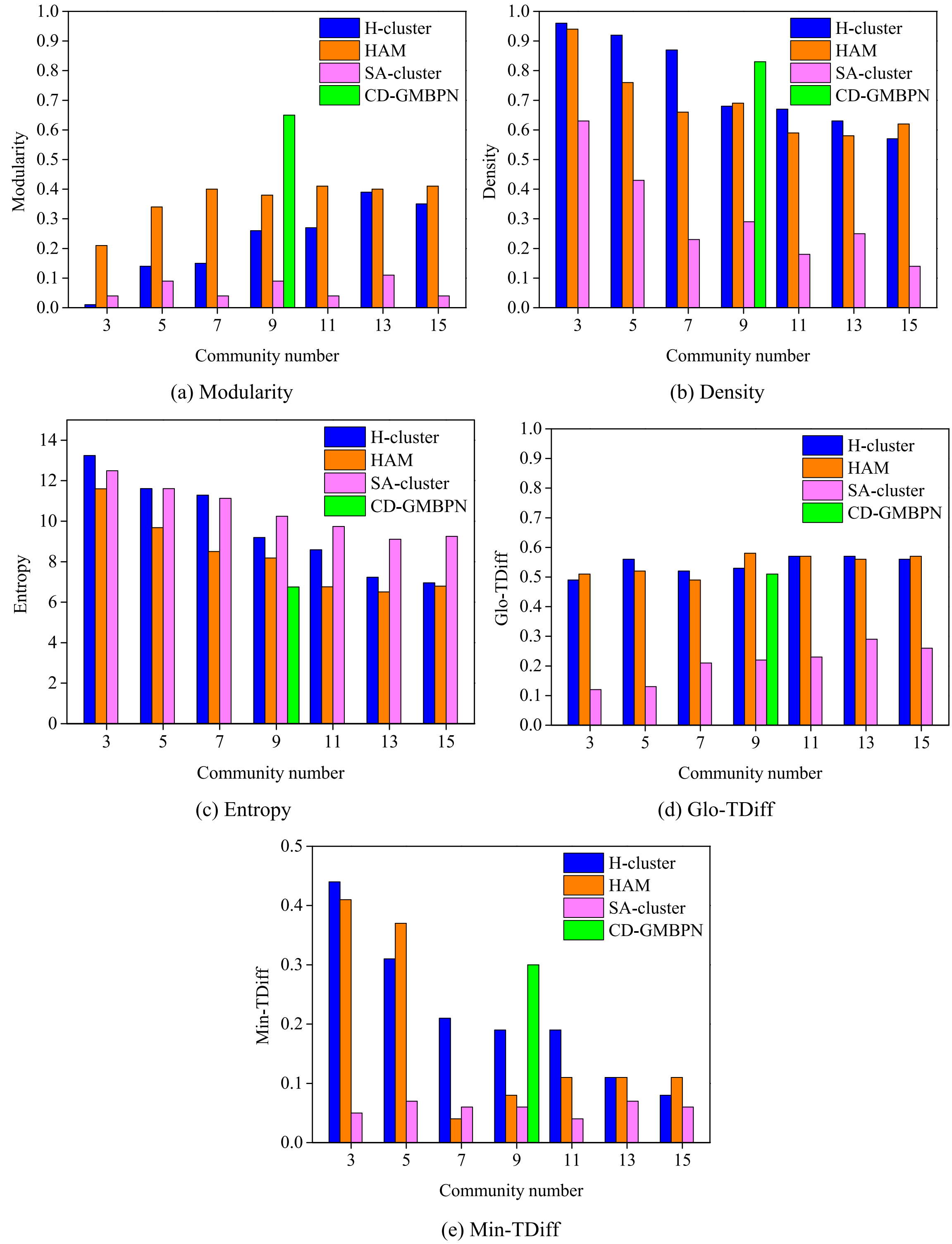


Fig. 6. Community detection results.

$$Entropy(\{C_j\}_{j=1}^{k}) = \sum_{i=1}^{n}\sum_{j=1}^{k}\frac{|C_j|}{|V|}entropy(a_i, C_j)$$
$$entropy(a_i, C_j) = -p_{ij}\log_2 p_{ij} \tag{11}$$

where $|V|$ represents the total number of nodes; p_{ij} represents the proportion of nodes in the community C_i with the attribute value a_i; The smaller the value of the *Entropy*, the better the community detection algorithm.

(4) ***TDiff***

The *TDiff* can be calculated according to Eqs. (6)–(8).

The above four metrics can be divided into two categories. *Modularity* and *Density* are used to evaluate the community detection results from the perspective of network topology; *Entropy* and *TDiff* evaluate the community detection results from the perspective of node attribute.

The experimental environment is ThinkPad X240 CPU/4G/32bit/

Table 3
Fusion value obtained by different algorithms (the optimal value is bold and underlined).

Algorithm	Community number						
	3	5	7	9	11	13	15
H-cluster	0.51	0.53	0.45	–	0.53	0.56	0.50
HAM	0.64	0.70	0.49	–	0.58	0.57	0.58
SA-cluster	0.11	0.16	0.15	–	0.14	0.23	0.16
CD-GMBPN	–	–	–	**<u>0.87</u>**	–	–	–

Win7. All of the simulation experiments are completed by Matlab2014a. We set the initial parameters $\alpha = \beta = 1$.

We design three experiments. Experiment 1 is used to evaluate the sensitivity of initial parameters in the CD-GMBPN algorithm. Experiment 2 and experiment 3 are used to evaluate the effectiveness and superiority of the algorithm. To give comprehensive evaluation and comparison results, we assign a suitable weight to each metric to get a fusion value. The larger the value is, the better the performance of the algorithm is. Therefore, a weight assignment method combining objective information and subjective information (Xu & Da, 2002) is used. In the method, the entropy method is used to obtain objective weights and the analytic hierarchy process (AHP) method is used to obtain subjective weights. In this paper, when using the AHP method, all of the metrics are considered to be equally important.

5.1.1. Experiment 1

In the CD-GMBPN algorithm, there are three parameters (i.e., α, β, and $\Delta\omega$) that need to be initialized. Therefore, experiment 1 aims to test the effect of initial parameters on the community detection results. Firstly, we fixed $\alpha = \beta = 1$. Then we analyze the sensitivity of the initial parameters to the community detection results by adjusting the weight increment $\Delta\omega$ (the value is from 0.05 to 1). Fig. 5 shows the change of evaluation indicators at different values of $\Delta\omega$. It can be seen from Fig. 5 that the choice of initial parameters will have a certain impact on the community detection results, but the overall fluctuation is acceptable and the results are relatively stable. By comprehensive analysis, it suggests that when the value of $\Delta\omega$ is between 0.4 and 0.55, the algorithm can achieve better results. Therefore, $\Delta\omega \in [0.4, 0.55]$ can be regarded as an optimal range, and the number of communities is 9 (as shown in Fig. 5(f)). We set $\Delta\omega = 0.45$ in the experiment 2 and experiment 3.

5.1.2. Experiment 2

Experiment 2 compares the FN algorithm, SAC algorithm, Louvain algorithm with our algorithm. All of the four algorithms do not need to provide the number of community in advance. Table 2 shows the values of the metrics obtained by the four algorithms. Since *Density* and *Entropy* only make sense when the community number is the same, therefore, experiment 2 only talks about *Modularity* and *TDiff*.

As shown in Table 2, the fusion value of the CD-GMBPN algorithm is the largest. It means the performance of our algorithm is the best. Specifically speaking, the Louvain algorithm and FN algorithm have the largest number of communities which is 21, while the number of communities obtained by SAC algorithm and our algorithm is 7 and 9 respectively. When comparing the metrics, the Louvain algorithm and FN algorithm obtain the optimal value of *Modularity* and *Glo-TDiff*, our algorithm is the second, and SAC algorithm is the last. However, for *Min-TDiff*, our algorithm is the best, the Louvain algorithm and the FN algorithm are the worst. It indicates that there are communities with high theme similarity. The reason is that the Louvain algorithm and FN algorithm only use the network topology information while ignore the node attribute information. For example, there are some isolated nodes in the network, but the themes of them are identical. These nodes should be clustered into the same community. Although both our algorithm and SAC algorithm consider the network topology information and node attribute information, all of the metrics obtained by our algorithm are better than the SAC algorithm. It is attributed to the adaptive weight optimization strategy. Moreover, it also avoids the disadvantage of determining the optimal weights in advance in our algorithm. In short, the proposed algorithm has advantages over the traditional complex network community detection algorithms.

5.1.3. Experiment 3

Experiment 3 compares our algorithm with the SA-Cluster algorithm, H-Cluster algorithm and HAM algorithm, and determines whether the number of communities (the number is 9) obtained by our algorithm is appropriate. It must be pointed out that the SA-Cluster algorithm, H-Cluster algorithm and HAM algorithm cannot automatically determine the optimal number of communities. The number of communities needs to be given in advance. Since it is impossible in this paper, algorithms are compared with different community numbers (3, 5, 7, 9, 11, 13, 15). The results are shown in Fig. 6.

Firstly, the results are discussed when the number of communities is 9. From the perspective of community structure, it can be seen from Fig. 6(a) that the *Modularity* obtained by our algorithm is optimal (0.65). It is much better than the other algorithms. The decomposition results have obvious community structure because our algorithm optimizes the *Modularity* while the other three algorithms do not. It can be seen from Fig. 6(b) that the *Density* obtained by our algorithm is about 0.83, which is in a best position. Therefore, by combining the results in Fig. 6(a) and (b), optimal community structure can be obtained by our algorithm. From the perspective of node attribute, the *Entropy* obtained by our algorithm is the optimal (6.75) in Fig. 6(c). And in Fig. 6(d), the

Table 4
Community detection results of the GMBPN of elevator design.

Community ID	Activity ID
1	3, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31
2	2, 12, 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, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 249
3	77, 78, 79, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 251
4	1, 80, 96, 97, 98, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 248, 267, 269, 270
5	205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245
6	246, 247
7	250, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 268
8	271, 272, 273, 274, 275
9	4, 5, 6, 7, 8, 9, 10, 11, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321

Glo-TDiff of the HAM algorithm and the H-cluster algorithm is slightly better than our algorithm. In Fig. 6(e), the *Min-TDiff* of our algorithm is 0.30, which is much larger than the other algorithms. Therefore, the theme of the obtained communities by our algorithm is better than others'. In conclusion, the metrics of our algorithm are optimal or sub-optimal, which indicates that the proposed algorithm can get better results compared with other three algorithms. Moreover, the fusion value of H-cluster, HAM, SA-cluster and CD-GMBPN is 0.71, 0.72, 0.35 and 0.96 respectively. In this regard, the performance of our algorithm is the best.

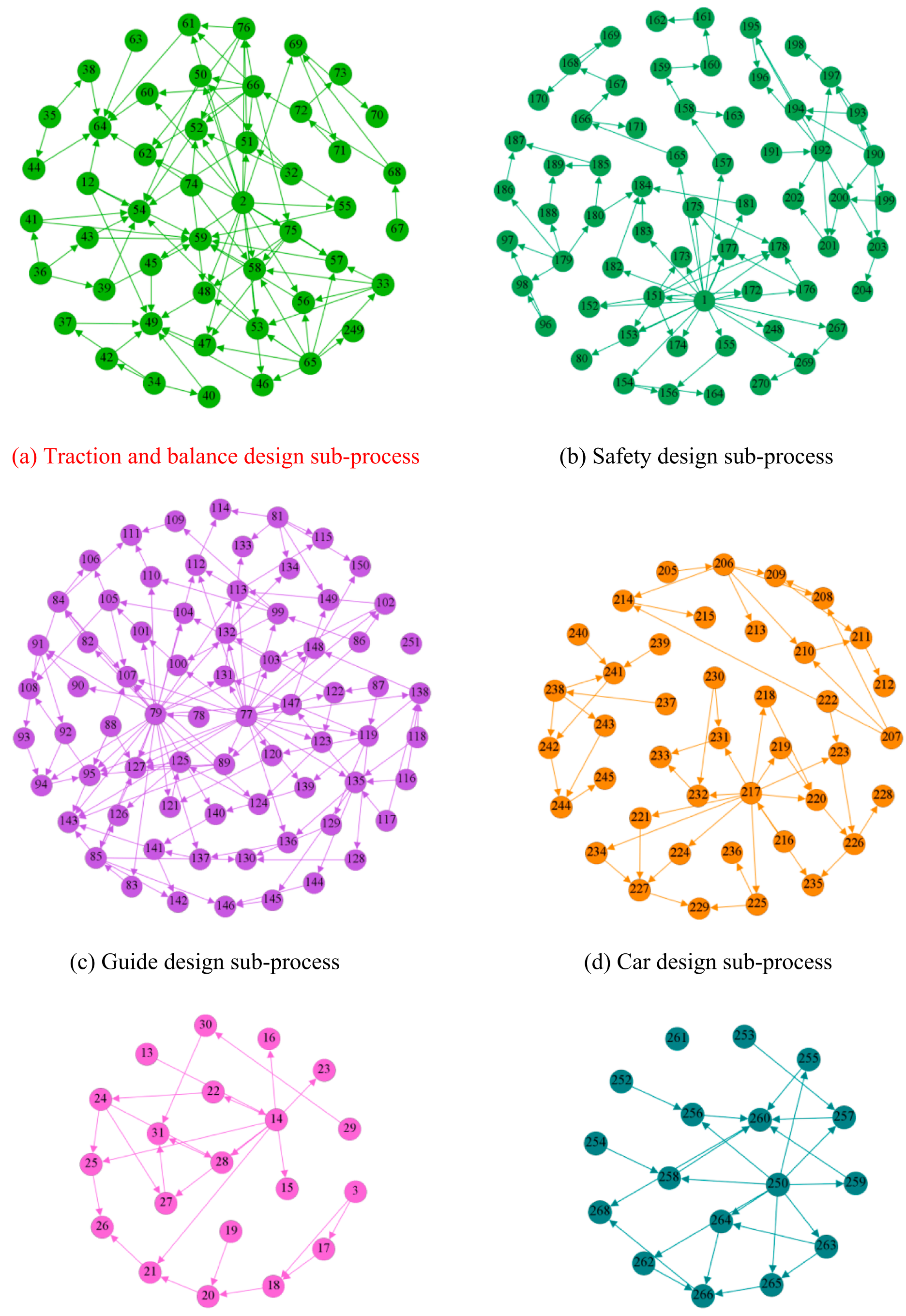
Secondly, the community detection results of each algorithm with different community numbers (3, 5, 7, 11, 13, 15) are compared with the results obtained by our algorithm. Like experiment 2, we only talks about *Modularity* and *TDiff*. The fusion values are shown in Table 3.

As shown in table 3, the fusion value of the CD-GMBPN algorithm is much better than other algorithms with different community numbers.

In conclusion, it is believed that the proposed algorithm can get the appropriate number of communities.

5.2. Decomposition results

The above experiments have proved the effectiveness of the community detection algorithm (the first phase of the GMBPN decomposition). Based on the proposed algorithm, the community detection



(a) Traction and balance design sub-process

(b) Safety design sub-process

(c) Guide design sub-process

(d) Car design sub-process

(e) Wire rope design sub-process

(f) Hoist-way design sub-process

Fig. 7. GMBPN decomposition results.

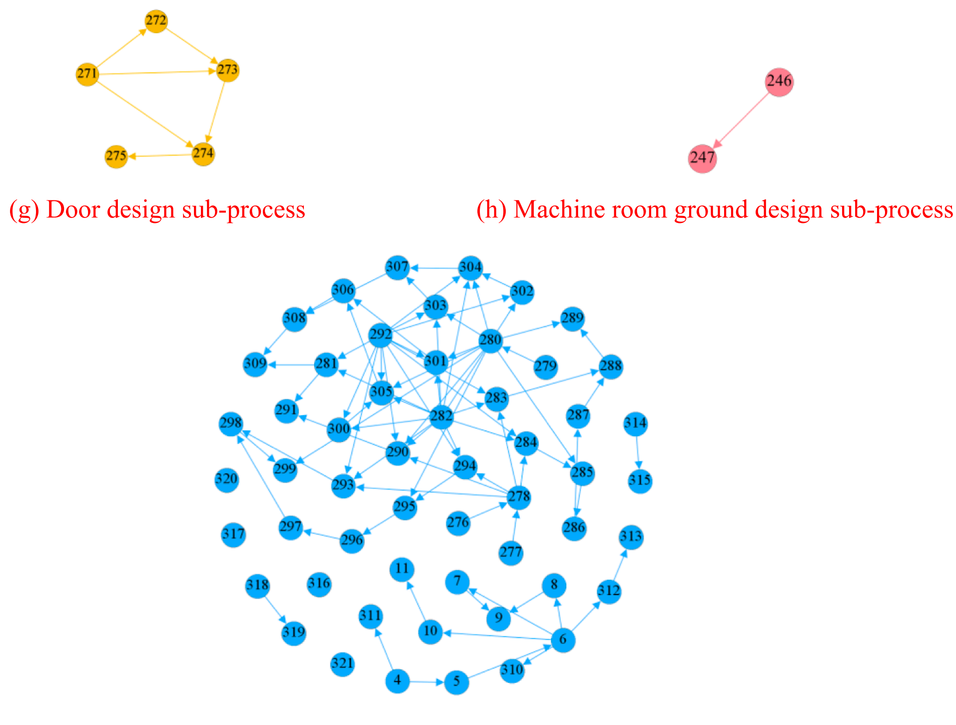

(g) Door design sub-process
(h) Machine room ground design sub-process
(i) Power control design sub-process

Fig. 7. (*continued*)

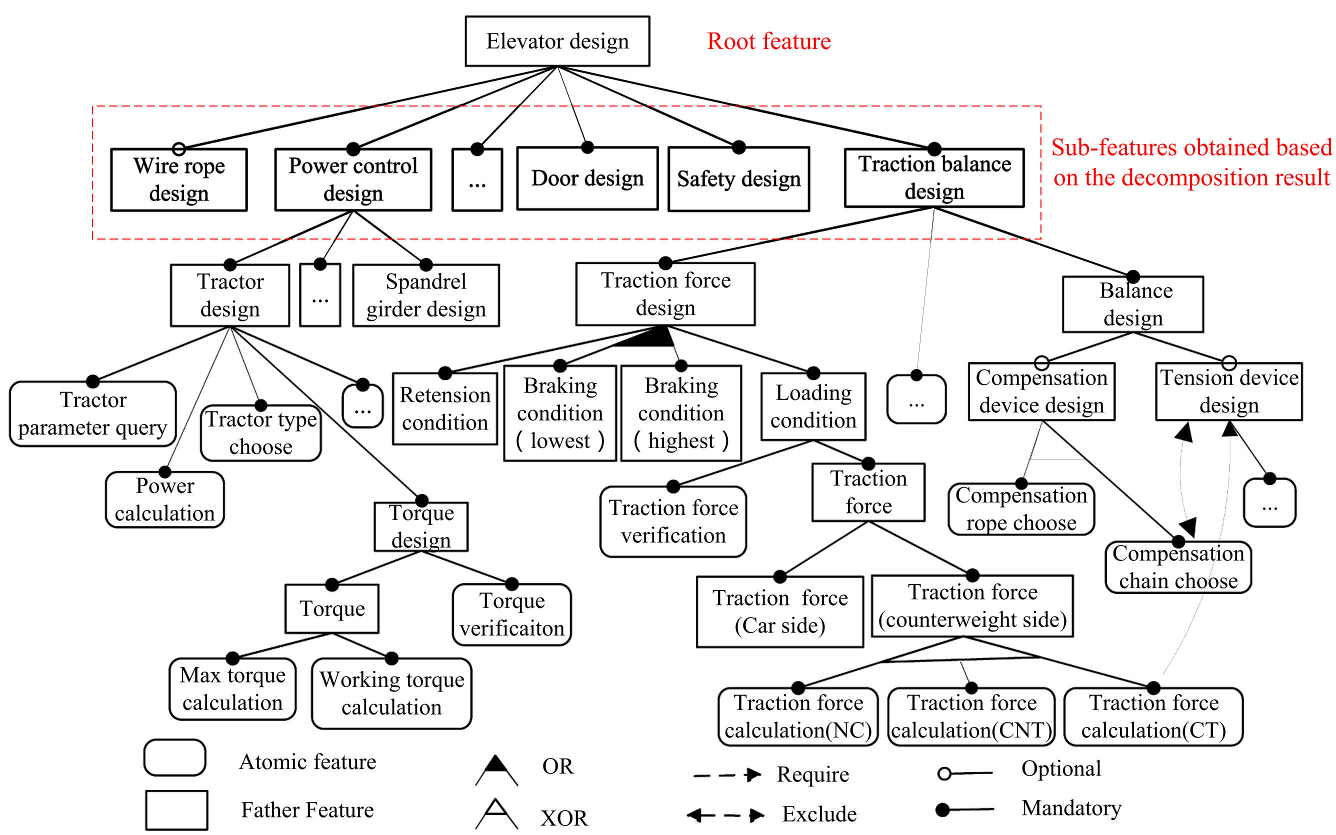


Fig. 8. Global feature model of elevator design.

results are shown in Table 4.

Based on the community detection results in Table 4, nine sub-processes are reconstructed (the second phase of the GMBPN decomposition). The final GMBPN decomposition results are shown in Fig. 7.

5.3. Discussion

To reduce complexity in analyzing the global manufacturing business process, a complex manufacturing business process decomposition approach was proposed. Based on the above experiments and results, the proposed approach was validated.

Compared with the six classical algorithms, the results in the experiments indicate that the community detection algorithm of the GMBPN is more effective than other classical algorithms. Our algorithm has the following advantages: firstly, the algorithm can automatically determine the optimal number of communities without prior giving; secondly, the weights of the network topology information and node attribute information are not necessary to be given in advance, and can be dynamically optimized in the community detection process. As our algorithm introduces the dynamic adjustment strategy to optimize the weights, the computational time of our algorithm is relatively larger than other algorithms. The dynamic adjustment strategy takes up most of the computational time. In fact, our algorithm mainly concentrates on the accuracy and effectiveness of the decomposition results which can help domain experts save a lot of time (several days). In this respect, more computational time is necessary to obtain better results.

Based on community detection results, the nine sub-processes are obtained. These sub-processes express the sub-functions of the whole elevator design business process. Actually, when analyzing the results (as shown in Fig. 7), the sub-processes are related to different themes about “traction balance design”, “safety design”, “guide design”, “car design”, “wire rope design”, “hoist-way design”, “door design”, “machine room ground design” and “power control design” respectively. In other words, the results suggest practical significance.

According to the decomposition results of the GMBPN of elevator design, the sub-features (i.e., wire rope design, door design, traction balance design, etc.) of the elevator design (root feature) can be obtained directly. By decomposing these sub-features gradually, the global feature model can be constructed. The global feature model is shown in Fig. 8. Compared with the traditional method, our approach can improve the automation degree and construction efficiency of the global feature model. At the same time, because the decomposition results are in line with the actual situation, the constructed global feature model is more scientific.

Therefore, the proposed approach can provide theoretical support for complex manufacturing business process decomposition and help the domain experts to understand the global manufacturing business process much easier. It can be used as a supplementary method for domain experts to improve the efficiency and scientificness in the construction of feature model.

However, there are also some limitations. Firstly, the node attribute information may be not accurate because it is described by the domain experts. Therefore, the robustness of the community detection algorithm is worth exploring. Secondly, the complexity of obtained sub-processes is different. For example, compared with the door design sub-process (as shown in Fig. 7(g)), the guide design sub-process (as shown in Fig. 7(c)) is relatively complex. So how to further reduce the complexity of such sub-processes needs to be considered in the future study.

6. Conclusions

To enhance the understandability of the global manufacturing business process and reduce the difficulties in constructing the feature model, a novel complex manufacturing business process decomposition approach in cloud manufacturing is proposed in this paper. The proposed approach is evaluated and proved to be effective in the elevator design business process decomposition. The main contributions in this paper are as follows:

(1) A domain-oriented customization of complex manufacturing business processes in cloud manufacturing is presented. It contains three phases: global manufacturing business process modeling, feature model construction and personalized manufacturing business process customization.
(2) A global manufacturing business process network (GMBPN) is constructed. It is a domain-oriented complex network where nodes represent manufacturing business activities and edges represent dependency relationships of manufacturing business activities.
(3) A two-phase decomposition algorithm for GMBPN is proposed. In the community detection phase, the nodes in GMBPN with functional similarity are clustered into the same community with an adaptive strategy. The weights of network topology information and node attribute information can be dynamically adjusted to obtain the optimal community detection results. In the sub-process reconstruction phase, the corresponding sub-process is reconstructed in each community.

According to the preliminary discussion in this paper, our future research work is mainly carried out from the following aspects: firstly, the robustness of the algorithm would be researched when the node attribute information is not accurate; secondly, the decomposition algorithm would be improved to satisfy the further decomposition requirements of the complex sub-processes.

Acknowledgement

The authors gratefully acknowledge the support by the National Natural Science Foundation of China under Grant 61976193, Zhejiang Special Science and Technology Special Project (2014C01048, 2018C01064 and 2018C04020), Public Projects of Zhejiang Province (2017C31014), and Zhejiang Natural Science Foundation (LY19F020034).

References

Asadi, M., Mohabbati, B., Gröner, G., & Gasevic, D. (2014). Development and validation of customized process models. *Journal of Systems and Software, 96*(4), 73–92.
Batory, D. (2005). Feature models, grammars, and propositional formulas. *International conference on software product lines* (pp. 7–20). Springer-Verlag.
Blondel, V. D., Guillaume, J. L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics, 10*, 155–168.
Cheng, Y., Bi, L. N., Tao, F., & Ji, P. (2018). Hypernetwork-based manufacturing service scheduling for distributed and collaborative manufacturing operations towards smart manufacturing. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-018-1417-8.
Cheng, Y., Tao, F., Xu, L. D., & Zhao, D. M. (2018). Advanced manufacturing systems: Supply–demand matching of manufacturing resource based on complex networks and Internet of Things. *Enterprise Information Systems, 12*(7), 1–18.
Cheng, Y., Tao, F., Zhao, D. M., & Zhang, L. (2017). Modeling of manufacturing service supply–demand matching hypernetwork in service-oriented manufacturing systems. *Robotics and Computer Integrated Manufacturing, 45*(6), 59–72.
Cheng, H., Zhou, Y., & Jeffery, X. Y. (2011). Clustering large attributed graphs: A balance between structural and attribute similarities. *ACM Transactions on Knowledge Discovery from Data, 5*(2), 1–33.
Clempner, J. (2010). A hierarchical decomposition of decision process petri nets for modeling complex systems. *International Journal of Applied Mathematics and Computer Science, 20*(2), 349–366.
Dikici, A., Turetken, O., & Demirors, O. (2018). Factors influencing the understandability of process models: A systematic literature review. *Information and Software Technology, 93*, 112–129.
Ding, K., Jiang, P., Leng, J., & Cao, W. (2016). Modeling and analyzing of an enterprise relationship network in the context of social manufacturing. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 230*(4), 752–769.
ESPRIT Consortium AMICE (1993). *CIMOSA: Open system architecture for CIM* (2th ed.). Berlin: Springer-Verlag.
Geng, C., Qu, S. Y., Xiao, Y. Y., Wang, M., Shi, G. Q., Lin, T. Y., et al. (2018). Diffusion mechanism simulation of cloud manufacturing complex network based on cooperative game theory. *Journal of Systems Engineering and Electronics, 29*(2), 321–335.
Johannsen, F., & Leist, S. (2012). Wand and Weber’s decomposition model in the context of business process modeling. *Web Intelligence, 4*(5), 271–286.
Kusiak, A., & Wang, J. (1993). Decomposition of the design process. *Journal of Mechanical Design, 115*(4), 687–695.
Li, Y. F., Tao, F., Cheng, Y., Zhang, X. Z., & Ne, A. Y. C. (2016). Complex networks in advanced manufacturing systems. *Journal of Manufacturing Systems, 43*(3), 409–421.
Li, B. H., Zhang, L., Ren, L., Chai, X. D., Tao, F., Luo, Y. L., et al. (2011). Further discussion on cloud manufacturing. *Computer Integrated Manufacturing System, 17*(3), 449–457.
Li, B. H., Zhang, L., Wang, S. L., Tao, F., Cao, J. W., Jiang, X. D., et al. (2010). Cloud manufactuirng: A new service-oriented networked manufactuirng model. *Computer Integrated Manufacturing System, 16*(1) 1–7 + 16.
Li, W. X., Zhu, C. S., Wei, X., Rodrigues, J. J. P. C., & Wang, K. (2016). Characteristics

analysis and optimization design of entities collaboration for cloud manufacturing. *Concurrency and Computation: Practice and Experience, 29*(14), e3948.

Lin, T. Y., Yang, C., Gu, M., Li, B. H., Xiao, Y. Y., Shi, G. Q., et al. (2016). Application technology of cloud manufacturing for aerospace complex products. *Computer Integrated Manufacturing System, 22*(04), 884–898.

Liu, Y. K., Zhang, L., Tao, F., & Wang, L. (2013). *Development and implementation of cloud manufacturing: An evolutionary perspective. Proceedings of the Asme 8th international manufacturing science and engineering conference, Madison, Wisconsin* V002T02A007.

Ljubicic, M., Ivezic, N., Kulvatunyuo, S., Nieman, S., Anicic, N., & Marjanovic, Z. (2017). *Business process model life-cycle management in cloud manufacturing. Proceedings of the ASME 2017 12th international manufacturing science and engineering conference, Los Angeles, USA* V003T04A039.

Milani, F., Dumas, M., Matulevičius, R., Ahmed, N., & Kasela, S. (2016). Criteria and heuristics for business process model decomposition. *Business and Information Systems Engineering, 58*(1), 7–17.

Newman, M. E. J. (2004). Fast algorithm for detecting community structure in networks. *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics, 69*(6 Pt 2), 066133.

Newman, M. E., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E Statistical Nonlinear and Soft Matter Physics, 69*(2), 026113.

Pan, T., & Zheng, L. (2012). Trust network modeling for virtual enterprise cloud manufacturing. *International Journal of Digital Content Technology and Its Applications, 6*(5), 115–123.

Reijers, H. A., Mendling, J., & Dijkman, R. M. (2011). Human and automatic modularizations of process models to enhance their comprehension. *Information Systems, 36*(5), 881–897.

Ren, M. L., Ren, L., & Jain, H. (2018). Manufacturing service composition model based on synergy effect: A social network analysis approach. *Applied Soft Computing, 70*, 288–300.

Schulte, S., Hoenisch, P., Hochreiner, C., Dustdar, S., Klusch, M., & Schuller, D. (2014). Towards process support in cloud manufacturing. *Enterprise distributed object computing conference, Ulm, Germany* (pp. 142–149). .

Smirnov, S., Dijkman, R., Mendling, J., & Weske, M. (2010). Meronymy-based aggregation of activities in business process models. *Lecture Notes in Computer Science, 6412*, 1–14.

Song, T. X., Zhang, C. L., Li, C. H., & Huang, B. Q. (2013). Cloud manufacturing service platform for small and medium enterprises. *Computer Integrated Manufacturing System, 19*(05), 1147–1154.

Tang, A., & Viennet, E. (2012). Community detection based on structural and attribute similarities. *International conference on the digital society, Valencia, Spain* (pp. 7–12). .

Tao, F., Cheng, J. F., Cheng, Y., Gu, S. X., Zheng, T. Y., & Yang, H. (2017). SDMSim: A manufacturing service supply–demand matching simulator under cloud environment. *Robotics and Computer Integrated Manufacturing, 45*(6), 34–46.

Tao, F., Zhang, L., Venkatesh, V. C., Luo, Y., & Cheng, Y. (2011). Cloud manufacturing: A computing and service-oriented manufacturing model. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 225*(10), 1969–1976.

Van der Aalst, W. M. P. (2012). Distributed process discovery and conformance checking. *International conference on fundamental approaches to software engineering, Tallinn, Estonia* (pp. 1–25). .

Van der Aalst, W. M. P. (2013). Decomposing petri nets for process mining: A generic approach. *Distributed and Parallel Databases, 31*(4), 471–507.

Vanhatalo, J., Völzer, H., & Koehler, J. (2009). The refined process structure tree. *Data and Knowledge Engineering, 68*(9), 793–818.

Wiśniewski, P., Borys, M., Czyż, Z., Falkowicz, K., Kujawska, J., Kulisz, M., & Szala, M. (2017). Decomposition of business process models into reusable sub-diagrams. *International conference of computational methods in engineering science, Lublin, Poland* (pp. 01002). .

Wu, D., Thames, J. L., Rosen, D. W., & Schaefer, D. (2012). Towards a cloud-based design and manufacturing paradigm: Looking backward, looking forward. *ASME 2012 international design engineering technical conferences and computers and information in engineering conference, Chicago, USA* (pp. 315–328). .

Wu, L., He, M. G., & Han, Y. Y. (2014). Hypergraph clustering-based cloud manufacturing service management method. *Proceedings of the 2014 IEEE 18th international conference on computer supported cooperative work in design* (pp. 220–225). .

Wu, D., Schaefer, D., & Rosen, D. W. (2013). Cloud-based design and manufacturing systems: A social network analysis. *Proceedings of the international conference on engineering design, Seoul, Korea* (pp. 149–158). .

Xu, X. (2012). From cloud computing to cloud manufacturing. *Robotics and Computer Integrated Manufacturing, 28*(1), 75–86.

Xu, Z. S., & Da, Q. L. (2002). Study on method of combination weighting. *Chinese Journal of Management Science, 02*, 85–88.

Yao, X. F., Jin, H., Li, B., Zhang, J., Lin, Y., & Zhou, H. F. (2013). Event-driven service-oriented architecture for cloud manufacturing and its implementation with open source tool. *Computer Integrated Manufacturing System, 19*(03), 654–661.

Yao, X. F., Lian, Z. T., Li, Y. X., Jin, H., Xu, C., Tan, W., et al. (2012). Service-oriented architecture and integrated development environment for cloud manufacturing. *Computer Integrated Manufacturing System, 18*(10), 2312–2322.

Zhang, L., Luo, Y. L., Tai, F., Li, B. H., Ren, L., Zhang, X. S., et al. (2014). Cloud manufacturing: A new manufacturing paradigm. *Enterprise Information Systems, 8*(2), 167–187.

Zhang, Z. J., Zhang, Y. M., Lu, J. W., Xu, X. S., Gao, F., & Xiao, G. (2018). CMfgIA: A cloud manufacturing application mode for industry alliance. *The International Journal of Advanced Manufacturing Technology, 98*(9–12), 2967–2985.

Zhong, R. Y., Xu, X., Klotz, E., & Newman, S. T. (2017). Intelligent manufacturing in the context of industry 4.0: A review. *Engineering, 3*(5), 616–630.