

# A heuristic configuration solving process planning method for mechanical product configuration by imitating the crystal crystallization process

**Shuyou Zhang**

Zhejiang University

**Wenqi Ge**

Zhejiang University

**zili wang** (✉ [ziliwang@zju.edu.cn](mailto:ziliwang@zju.edu.cn))

Zhejiang University <https://orcid.org/0000-0002-4765-4410>

**Lemiao Qiu**

Zhejiang University

**Huifang Zhou**

Zhejiang University

---

## Research Article

**Keywords:** Mechanical product configuration, coupling strength, frequent itemset mining, configuration solving process planning

**Posted Date:** February 12th, 2021

**DOI:** <https://doi.org/10.21203/rs.3.rs-212268/v1>

**License:**  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

---

# **A heuristic configuration solving process planning method for mechanical product configuration by imitating the crystal crystallization process**

Zhang Shuyou<sup>1</sup> Ge Wenqi<sup>1</sup> Wang Zili<sup>1\*</sup> Qiu Lemiao<sup>1</sup> Zhou Huifang<sup>1</sup>

*(1.State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, Hangzhou, P.R. China)*

*Correspondence should be addressed to Wang Zili; ziliwang@zju.edu.cn*

**Abstract:** Configuration design is a process of composing customized product from a set of predefined component types with a set of well-defined rules which decide how items can be selected and combined. The configuration solving process is to instantiate all component types into instances. However, the configurator faces a problem of how to determine the instantiation order of component types in the configuration solving process. An unreasonable instantiation order may reduce the accuracy of the configuration result and extend the time required to finish the configuration. In response to the above problem, a heuristic configuration solving process planning method by imitating the crystal crystallization process (CCP) was proposed. According to the product configuration rules, the frequent itemset mining algorithm was used to mine the correlation between the attribute parameters of product configuration unit. By imitating the CCP and utilizing the calculated coupling strength, the configurator will determine a suitable instantiation sequence for component types. This method has been applied in the GB10 elevator produced by Canny Elevator. It is found that the configuration accuracy has been improved and the consumption of time has been reduced.

**Keywords :** Mechanical product configuration; coupling strength; frequent itemset mining; configuration solving process planning

## **1. Introduction**

With the fast improvement of people's living standards, enterprises have received more personalized demands from customers. Meanwhile, technological progress has promoted the renewal of products, which requires enterprises to produce various products at relatively low costs. Confronting with this pair of contradictory requirements, a great number of enterprises have gradually transformed their production patterns from mass production to mass customization. Configuration design is one of the effective ways to achieve mass customization. It can help enterprises to reduce the cost of research and development, and meet the demands of customers to the greatest extent [1].

The goal of configuration design is to compose customized product from a set of predefined component types with a set of well-defined rules which decide how items can be selected and combined [2,3]. Component types are characterized by attributes and represent sets of alternative instances. For example, the motor produced by a certain manufacturer has two types: motor-A and motor-B. Meanwhile, motor has attributes like rated voltage, rated power, etc. A motor can be instantiated to be motor-A or motor-B. The final result of the configuration design is a list of instances which are obtained by instantiating component types. The instantiation process of component types is just the configuration solving process. In order to better introduce the

configuration solving process, the concepts of product configuration unit (PCU) and product configuration template (PCT) are defined here. Each PCU is defined as a package which encapsulates a certain component type and its corresponding rules which are pre-defined. The rules are represented in an if-then style and determine which instance of the component type to be selected. PCT is an abstract object of a product series and it represents products that have similar shapes and functions. It is composed of PCUs and has a classical tree shape. According to the demand of customers, different PCUs can be selected to form various PCTs. The configuration solving process is to instantiate all the PCUs in the PCT into instances according to the rules and finally obtain a tree composed of instances (PCU instantiation is equivalent to the instantiation of component types).

When traditional machinery manufacturers like Canny Elevator develop configurators, they usually traverse the PCT tree from top to bottom and instantiate the PCU in the order of nodes. During the process of instantiation, the consequence of configuration solving process can be greatly affected by the constraints between PCUs. The constraints between PCUs indicate the situation where the instance of a component type in a PCU affects the instantiation of other PCUs. For example, elevator product contains a traction machine and a driving sheave. The instantiation of the traction wheel will be affected by the instance of the traction machine. Sometimes we also refer this relationship between PCUs as coupling and use the concept of coupling strength to express the closeness of the relationship between the two. Obviously, if the driving sheave is instantiated before the traction machine, we will not get the instantiation result of driving sheave. The mismatch of the driving sheave can be avoided by arranging a more reasonable PCU instantiation sequence for the configuration solving process in advance.

Arranging a reasonable PCU instantiation sequence is the content of the configuration solving process planning in this paper. A method is proposed to achieve this object. The entire problem-solving process is divided into two steps by us: coupling strength quantification and instantiation sequence arrangement. Coupling strength quantification is a measure of the strength of the constraints between PCUs, it is the forerunner of instantiation sequence arrangement. For the measurement of coupling strength in this problem, most of the existing methods which can be used are vague and subjective such as triangular fuzzy numbers. There is no objective quantitative indicator at present. In this case, we start from the data itself and introduce data mining into this process. The amount of information contained in each parameter from configuration rules is mined and the degree of information dependence between each PCU is used as a measure of coupling strength. After clarifying the constraint relationship through the coupling strength, instantiation sequence arrangement can be implemented. The Design Structure Matrix (DSM) proposed by Steward [4] can be used here. However, the DSM matrix has the disadvantage of not being flexible enough to deal with larger problems. Thus, we propose a heuristic method by imitating the crystal crystallization process (CCP). This method is more flexible when the product structure is more complicated and the number of PCUs is large.

The remainder of the paper is organized as follows. Section 2 is the literature review. Section

3 briefly introduces the concept of product configuration unit (PCU) and expresses the contributing factor of coupling. The calculation method of coupling strength is given in Section 4. Section 5 presents our method for arranging the PCU instantiation sequence by imitating the CCP. In Section 6 an industrial illustrative example of configuration design is illustrated and some experiments of comparison are carried out. Concluding remarks are drawn in Section 7.

## **2. Literature review**

### *2.1. Knowledge representation and reasoning method in configuration design*

The representation of knowledge is a key issue in product configuration design. Knowledge reasoning method is the basis of configuration solving process. One of the earliest configuration systems is R1/XCON, which is initially implemented by John McDermott on the basis of OPS5 [5,6]. R1/XCON is a rule-based configurator which operates on a working memory that stores assertions made by rules. Rules are represented in an if-then style. If a rule is activated (given that the “if” condition is fulfilled) it can modify the contents of the working memory [7]. The rule-based system intermingles domain knowledge and problem solving knowledge together. As for improvement, the definition of model-based configuration which strictly separates domain knowledge and problem solving knowledge was proposed [8,7]. A large variety of theories have been proposed later under this framework to express knowledge. Mailharro and Amilhastre et al. [9,10] treat the product configuration as a constraint satisfaction problem. Wang [11] introduces a case retrieval method for machine tool configuration. Zeng [12] uses a product model and configuration rules to represent knowledge. Jangyong Lee [13] uses the functional model to build a product configuration model and configure the train bogie. Dong et al.[14] utilize ontology web language and knowledge graph to model configuration knowledge. The usage of ontology can improve the accessibility of configuration knowledge. The knowledge representation and reasoning method used in this paper belongs to the model-based configuration. PCU and PCT decompose products into a hierarchical structure, representing hierarchical domain knowledge. Problem solving knowledge is expressed in the form of configuration rules. The cooperation of PCU, PCT and configuration rules makes knowledge easier to maintain and manage.

### *2.2. Coupling strength quantification method*

Coupling strength quantification is the forerunner of instantiation sequence arrangement. The instantiation process of one PCU is similar to a small design activity. Thus, we assume that the process of PCU instantiation is a small design activity of the overall configuration solving process. These small activities cooperate with each other and deliver information to complete the entire configuration solving process. Some scholars have conducted quantitative research on the dependence between design activities. Yassine et al. [15] set up a subjective evaluation scale in the way of verbal descriptions to measure the degree of coupling between design activities. Srour et al.

[16] evaluated the interdependence of projects by consulting professionals. Lin et al. [17] assessed the degree of dependency between activities in the manner of interviewing experts and used triangular fuzzy numbers to digitize it. All these methods cannot be separated from human subjective evaluation. The activities of instantiation process in our work is different from them in that the PCUs have certain parameters and rules. These parameters and rules contain a lot of information and can be mined. Therefore we introduce data mining methods into the calculation process of coupling strength. Based on data mining, the human evaluation step has been removed, thus the objectivity of the coupling strength can be guaranteed.

### 2.3. *PCU instantiation sequence arrangement*

The PCU instantiation sequence arrangement is essentially the management and planning of activities. There are several ways to achieve this object. Critical Path Method (CPM) expresses the relationship between activities in the form of a directed graph, which was applied in the field of design activity management in an early period [18]. The Project Evaluation and Review Technique (PERT) method was proposed almost simultaneously with CPM. It is a digraph representation of a project where the nodes are arranged along a time line. Comparing with CPM, PERT has more advantages in flexibility [19]. When applying network analysis to manage design activities, the information feedback and the interaction between activities are often ignored. The Design Structure Matrix (DSM) proposed by Steward [4] which constructs a matrix to represent the interaction between elements effectively solves this problem. Different from network analysis, it is a more concise and compact representation method, and it is easier to operate and store by computer. Since then, designers have proposed various partitioning and tearing algorithms to manage and schematize design activities based on the design structure matrix. At the same time, various variants of the design structure matrix have emerged, such as the Work transformation matrix (WTM) [20], the two-dimensional Numerical DSM (NDSM) [15], Hierarchical design structure matrix (HDSM)[21], Design dependency matrix (DDM) [22,23]. Although these DSM-related methods have many conveniences, they still have a disadvantage that they are not efficient when dealing with large-scale problems. Therefore, this paper imitates the crystal crystallization process (CCP) and proposes a heuristic method. Unlike traditional DSM methods which operate the matrix as a whole, each activity is treated as a basic unit of operation in our method. When new design activity (the PCU which is needed to be instantiated) is added, only a few steps are required, which apparently improves the flexibility of the configuration and reduce the configuration time.

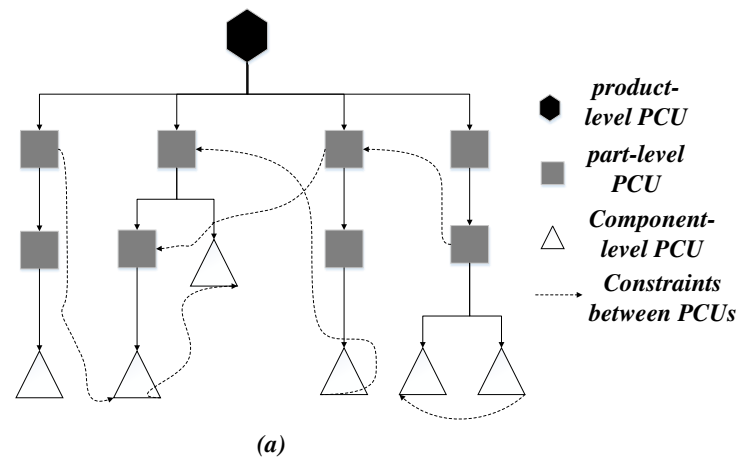
## 3. **Coupling characteristics of PCT**

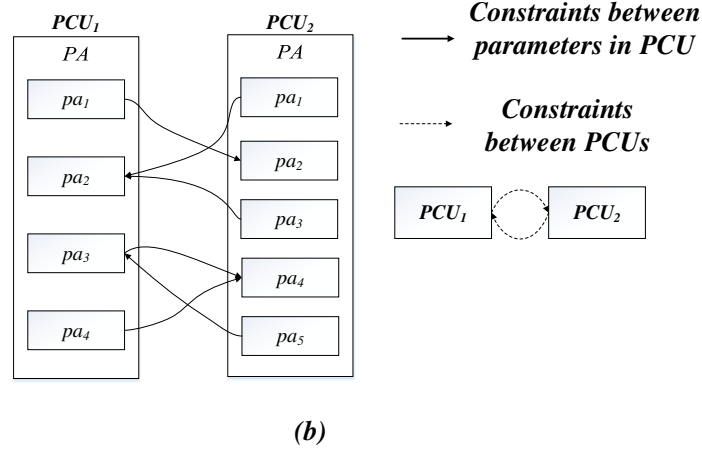
Being the basic unit of mechanical configuration design, the PCU is a set of components that decompose and encapsulate the function, principle, and structure of the mechanical product. Each PCU contains some knowledge required for configuration design including configuration rules, the instances of component types, configuration operations, attribute parameters, etc. It can be

expressed as  $PCU = \{ID, FA, PA, RU, LE, KEY, OP, INS\}$ , where ID is the unique identifier of the PCU, FA is the parent node of the PCU,  $PA = \{pa_1, pa_2, \dots, pa_m\}$  is a set of attribute parameters which describe the characteristics of the component type corresponding to the PCU,  $m \geq 0$ ,  $RU = \{ru_1, ru_2, \dots, ru_n\}$  is a set of configuration rules,  $n \geq 0$ , LE indicates whether the PCU is located at the leaf node of the configuration template structure tree, KEY indicates whether the PCU is a key configuration unit,  $OP = \{op_1, op_2, \dots, op_k\}$  represents a set of configuration operations on the PCU, including reading attribute parameters, matching configuration rules, finding instances, etc,  $INS = \{ins_1, ins_2, \dots, ins_k\}$  represents the set of alternative instances.

PCU is the basic unit that constitutes the PCT. The product family model of complex mechanical products is often hierarchical. Thus, the PCT has a conspicuous tree-type structure. A PCU can be decomposed into multiple sub-PCU units, and can also be the sub-unit of a PCU together with other PCU units. According to the position of PCU in PCT, all PCUs can be divided into three categories: part-level PCU, component-level PCU and product-level PCU. The product-level PCU is located at the top level of PCT, that is, the root node of the structure tree; the component-level PCU is located at the middle node of the structure tree; and the part-level PCU is located at the leaf node of the structure tree which cannot be further decomposed.

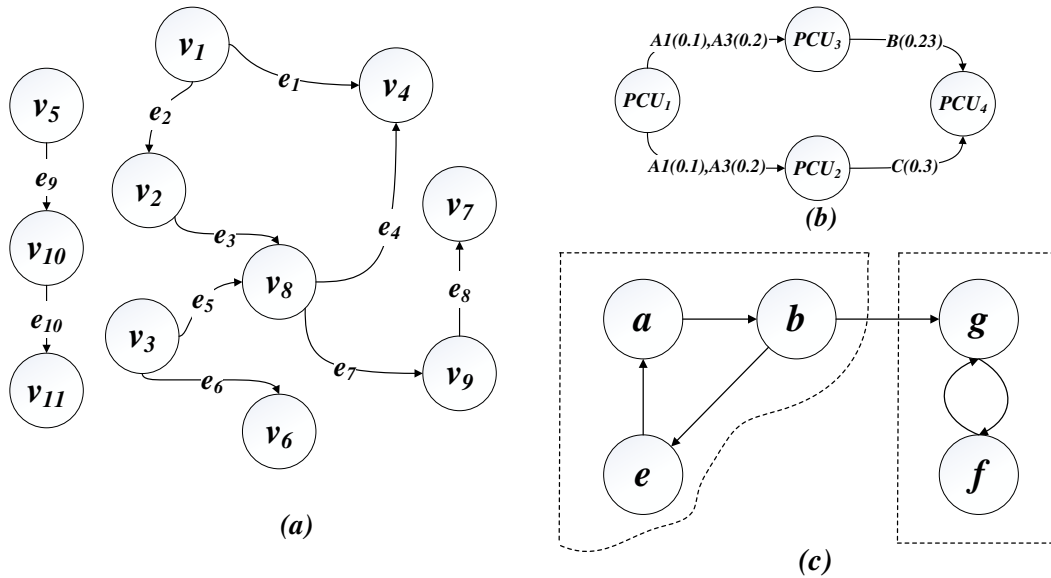
PCT and configuration parameters converted from customer demands are the input of the configurator. Configuration starts from the root node of PCT and executes from top to bottom sequentially. The configurator takes out the PCU in each node. The configuration rules are matched to instantiate each PCU until all levels of PCUs are instantiated. Finally, PCT is transformed into a tree composed of instances, which is the Bill of Material (BoM) of configuration design. The PCT structure tree is shown in Figure 1 (a).





**Fig. 1.** Constraints in PCT. (a) Product configuration template structure tree, (b) Cause of coupling between PCUs.

The dotted line in Figure 1 (a) represents the constraints between PCUs in PCT. The constraint of PCU mainly results from the constraints between the attribute parameters encapsulated in the PCU. As shown in Figure 1 (b), there are four parameters encapsulated in the attribute parameter set  $PA = \{pa_1, pa_2, pa_3, pa_4\}$  of PCU<sub>1</sub> and five parameters encapsulated in the attribute parameter set  $PA = \{pa_1, pa_2, pa_3, pa_4, pa_5\}$  of PCU<sub>2</sub>. Among them, PCU<sub>1</sub> affects PCU<sub>2</sub> by parameters  $pa_1, pa_3, pa_4$ ; and PCU<sub>2</sub> affects the parameters of PCU<sub>1</sub> by  $pa_1, pa_3, pa_5$ . This means that to complete the instantiation of PCU<sub>1</sub>, values of  $pa_1, pa_3, pa_5$  from PCU<sub>2</sub> is required and vice versa. The coupling relationship between PCUs can be represented by a directed graph, as shown in Figure 2. Let the directed graph to be  $B = \{V, E\}$ , where  $V = \{v_1, v_2, \dots, v_n\}$  is the set of the vertices that represents the collection of all PCUs,  $E = \{e_1, e_2, \dots, e_m\}$  is the set of edges that represents the constraints between PCUs.



**Fig. 2.** Directed graph of coupling relationship. (a) Common directed graph, (b) Directed graph

without strongly connected components, (c) Directed graph with strongly connected components.

As shown in Figure 2 (a), the vertices of this graph can be divided into two sets  $\{v_5, v_{10}, v_{11}\}$ ,  $\{v_1, v_2, v_3, v_4, v_6, v_7, v_8, v_9\}$ . They are two independent connected components in graph theory. We define them as two coupled PCU sets.

For a directed graph  $G$ , if for each pair of  $v_i, v_j \in V, v_i \neq v_j$ , there exists paths from  $v_i$  to  $v_j$  and from  $v_j$  to  $v_i$ , then  $G$  is a strongly connected graph. The maximally strongly connected subgraph in a directed graph is called a strongly connected component. In terms of the above definition, the directed graphs of the coupling relationship between PCUs are divided into two categories: directed graphs that do not contain strongly connected components, as shown in Figure 2 (b); and directed graphs that contain strongly connected components, as shown in Figure 2 (c). For the case of Figure 2 (b), the PCU instantiation sequence arrangement is relatively simple, and it can be set according to the direction of the constraint. Thus, the PCU instantiation sequence can be decided as  $PCU_1 \rightarrow PCU_2 / PCU_3 \rightarrow PCU_4$ . For the situation shown in Figure 2c, there is a bidirectional constraints between  $g, f$  and  $a, b, e$ . To determine the instantiation order for the PCU contained in the strongly connected component, a tearing algorithm is required.

#### 4. Coupling strength quantification of PCT

##### 4.1. Configuration rules

Configuration rules are one of the key information required in the configuration solving process. The configuration rule can be expressed as:

*Rule*:            < rule ID >  
*IF*                < premise clause > {AND < premise clause >}\*  
                          (\* indicates repeated 0 or more times)  
*THEN*            < conclusion clause >

Each PCU contains a series of configuration rules  $ru_1, \dots, ru_n$ , which is encapsulated in  $RU = \{ru_1, ru_2, \dots, ru_n\}$ . <premise clause> represents the conditions that the PCU is required to satisfy during the instantiation process, usually in the following two forms:

- 1、  $exist\ of\ < PCU > = < true > | < false >$
- 2、  $< comparator > < value >$

The parameters involved in the IF part of the configuration rules can be attribute parameters of other PCUs, or can be configuration parameters converted from customer demands. When the rule contains the attribute parameters of other PCUs, it means that the instantiation of current PCU requires the information passed from other PCUs, which also indicates that the PCU is constrained by other PCUs. The configuration parameters converted from demands have no influence on the constraints between PCUs, so they are not considered.



The <conclusion clause> part of the configuration rule is the corresponding instance code of the PCU after instantiating. When the PCU meets the IF part of the configuration rule, the PCU is instantiated as a specific instance.

#### 4.2. FP-growth algorithm for frequent item mining

Frequent itemset mining is usually used to discover the combination relationship between the elements in the data set. Let the itemset  $I = \{i_1, i_2, \dots, i_m\}$  be a collection of  $m$  different items, the transaction data set  $D = \{T_1, T_2, \dots, T_n\}$  represents a collection of  $n$  different transactions which satisfy  $T_k \subseteq I$ . Assume a set  $J = \{i_a, i_b, \dots, i_q\} \subseteq T_k$ . For set  $J$ ,  $count(J)$  is defined as its support, which is expressed as the frequency of its occurrence in the transaction data set  $D$ . The set  $J$  that meets the pre-set minimum support (min\_support) is defined as the frequent item set of the transaction data set  $D$ , which represents the combination of items that frequently appear in the data set  $D$ .

FP-growth algorithm is one of the most used frequent item mining algorithms. The execution of this algorithm can be divided into two stages. The first stage generates an FP-tree to store the transaction data set by traversing the database, and the second stage generates frequent itemsets by traversing the FP-tree. The basic steps of the algorithm are as follows:

**Step 1.** Generate the data set.

**Step 2.** Set the minimum support, remove non-frequent elements from the original data set and sort them in descending order in terms of the support of each element. After that, store the data in a list, and construct the header table according to the frequency of occurrences of each element.

**Step 3.** Set NULL as the root node and create FP-tree from top to bottom.

**Step 4.** Starting from the bottom of the item header table to detect the condition pattern base in turn. The frequent itemsets are mined according to the conditional pattern base.

Based on configuration rules, the FP-growth algorithm is exploited to mine frequent items of the attribute parameters encapsulated in the PCU, which is used as a measure of the amount of information contained in the attribute parameters. After that, a method is provided in this paper to measure the coupling strength between the PCUs.

#### 4.3. The process of coupling strength quantification

In this paper, the strength of information dependence between PCUs is used as a measure of the coupling strength. We believe that the parameters that appear more frequently in the rules contain more information. The implementation steps are as follows:

**Step 1** For  $PCU_i (i = 0, 1, \dots, n)$ , its attribute parameter set is  $PA_i = \{pa_{i1}, pa_{i2}, \dots, pa_{im}\}$ . According to the analysis in Section 3, some of the attribute parameters of  $PCU_i$  can restrict other  $PCUs$ , let these parameters to be  $\{pa_{i1}, pa_{i2}, \dots, pa_{il}\}, 0 \leq l \leq m$ . Counting the total emergence times of these parameters in the <prerequisite clause> part of all configuration rules, denoted as  $K$ , and counting the emergence times of each parameter, denoted as a set  $\{k_{i1}, k_{i2}, \dots, k_{il}\}$ , where  $k_{il}$

is the occurrence times of the parameter  $pa_{il}$ . Combining the parameter set and the emergence times sets to obtain a new set  $KPA_i = \{\{pa_{i1}, k_{i1}\}, \{pa_{i2}, k_{i2}\}, \dots, \{pa_{il}, k_{il}\}\} = \{A_{i1}, \dots, A_{il}\}$ .

**Step 2** Based on the <prerequisite clause> part of configuration rules, frequent items of parameter combination is mined. For each piece of configuration rule, the attribute parameters that belongs to  $PCU_i$  and are contained in the <prerequisite clause> are extracted to form a piece of *item*:  $\{pa_{ia}, pa_{ib}, \dots, pa_{iq}\}$ . All *items* make up the data set  $itemsets = \{item_1, item_2, \dots, item_p\}$ , where  $0 \leq p \leq rule\_sum$ , and  $rule\_sum$  is the number of configuration rules. FP-growth algorithm is used to mine frequent itemsets of parameter combinations of  $itemsets$ , getting  $frequent_s_i = \{U_{i1}, U_{i2}, \dots, U_{ig}\}$ . For  $frequent_s_i, U_{ij} = \{FRA_{ij}, su_{ij}\} (j = 1, 2, \dots, g)$  is a frequent item of parameter combination, where  $FRA_{ij} = \{pa_{ir}, \dots, pa_{is}\}$ , and  $su_{ij}$  is the support of the frequent item. The significance of frequent item mining is that when a combination of parameters in a frequent item appears, it can be considered that it contains more information than a single parameter.

**Step 3** Combine  $KPA_i, frequent_s_i, KPA_i \setminus frequent_s_i = \{U_{i1}, U_{i2}, \dots, U_{ig}, A_{i1}, \dots, A_{il}\}$ . Then, for each element in the set, calculate the weight by formula

$$\alpha_{U_{ik}} = \frac{su_{ik}}{\sum_{j=1}^g su_{ij} + \sum_{m=1}^l k_{im}} (k = 1, 2, \dots, g)$$

$$\alpha_{A_{ik}} = \frac{k_{ik}}{\sum_{j=1}^g su_{ij} + \sum_{m=1}^l k_{im}} (k = 1, 2, \dots, l)$$
(1)

**Step 4** For all PCUs, after calculating  $\alpha_{U_{ik}}$  and  $\alpha_{A_{ik}}$ , the coupling strength between each PCU is calculated in terms of the constraints between PCUs. As for  $PCU_i$  and  $PCU_j$ , if  $PCU_i$  directly or indirectly restricts  $PCU_j$  through attribute parameter set  $PA_i' = \{pa_{i1}', pa_{i2}', \dots, pa_{in}'\} \subseteq PA_i$ , then the coupling strength of  $PCU_i$  to  $PCU_j$  should be

$$coup(i, j) = \begin{cases} \sum_{k=1}^n \alpha_{A_{ik}} + \alpha_{U_{ig}} & , \exists PA_i' \subseteq PA_i \text{ such that } PA_i' = FRA_{ig}; \\ \sum_{k=1}^n \alpha_{A_{ik}} & , \text{else.} \end{cases}$$
(2)

$coup(j, i)$  can also be calculated in the same way. The larger value of  $coup(i, j)$  indicates that if  $PCU_j$  wants to complete the instantiation, it requires more information from  $PCU_i$  which indicates the stronger coupling.

Take Figure 2 (b) as an example to further explain the calculation method of coupling strength. As shown in the figure,  $PCU_1$  restricts  $PCU_3, PCU_2$  through parameters  $\{A_1, A_3\}, \{A_1, A_2\}$ , and the weights of  $A_1, A_2, A_3$  are 0.1, 0.3, 0.2. At the same time, we know that the attribute parameters of  $PCU_1$  exist frequent itemset  $U = \{A_1, A_2, 0.25\}$ . As the same,  $PCU_2, PCU_3$

restricts  $PCU_4$  through parameters. Thus, as for  $PCU_4$ , to complete the instantiation, it first requires the information of parameters  $\{A_1, A_2, A_3\}$  from  $PCU_1$ , helping  $PCU_2, PCU_3$  finish instantiation. Then, it needs the information of parameters  $B, C$  from  $PCU_2, PCU_3$ . Hence, as for  $PCU_4$ :

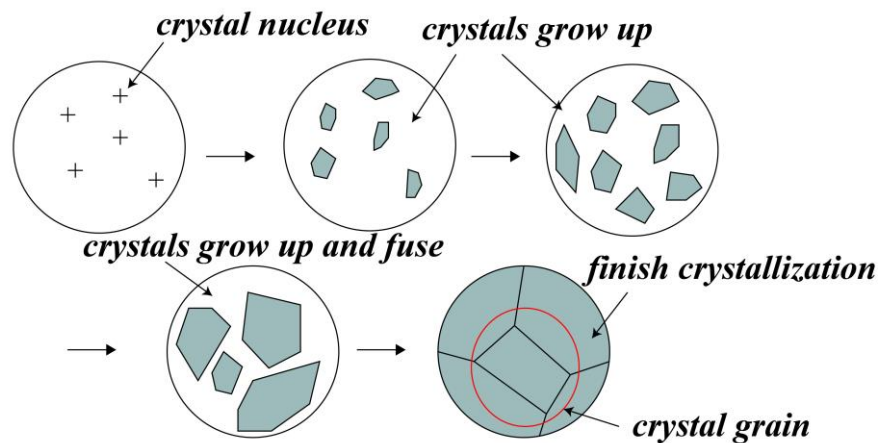
$$coup(1,4) = 0.1 + 0.3 + 0.2 + 0.25 = 0.85, \quad coup(2,4) = 0.3, \quad coup(3,4) = 0.23$$

$$\text{The same, } coup(1,3) = 0.1 + 0.2 = 0.3, \quad coup(1,4) = 0.1 + 0.3 + 0.25 = 0.65.$$

## 5. CCP imitated product configuration solving process planning method

### 5.1. Heuristic algorithm of the instantiation sequence arrangement

For a specific model of a complex mechanical product, the PCT usually contains hundreds or even thousands of PCUs. The coupling relationship between PCUs is very sophisticated. It requires us to make a reasonable plan for instantiation order of PCUs. Inspired by Jafari et al.[24], a heuristic algorithm which treats the arrangement process of PCU instantiation as a CCP is proposed in this paper for the coupled PCUs. The CCP in the nature is shown in Figure 5. As is illustrated in the figure, a big crystal is formed after crystallization, and the crystal is made up of several crystal grains. Crystal grains are made of atoms or molecules. Inside each crystal grain, the atoms and molecules are arranged regularly. In our paper, each PCU in the configuration template is regarded as an atom, and let all PCUs form a pool. By imitating the CCP in the nature and using the pre-calculated coupling strength, each PCU gradually precipitates to form crystal grains. All grains form a big crystal. Each grain in the crystal represents a coupled PCU set. The contents of the crystallization process are as follows.



**Fig. 3.** Process of crystallization phenomenon in the nature

#### (1) Crystal nucleation

This step simulates the nucleation process of crystal crystallization, and the main purpose is to select a PCU from the PCU pool to form a crystal nucleus as the starting point for crystal growth. To achieve this step, we assume that there are  $n$  PCUs in the configuration product. For  $PCU_i$  in the PCU pool, the atom diameter is defined to be

$$D_i = \frac{\sum_{j=1, j \neq i}^n coup(i, j) + (\sum_{i=1}^n \sum_{j=1}^n coup(i, j)) / n^2}{\sum_{j=1, j \neq i}^n coup(j, i) + (\sum_{i=1}^n \sum_{j=1}^n coup(i, j)) / n^2} \quad (3)$$

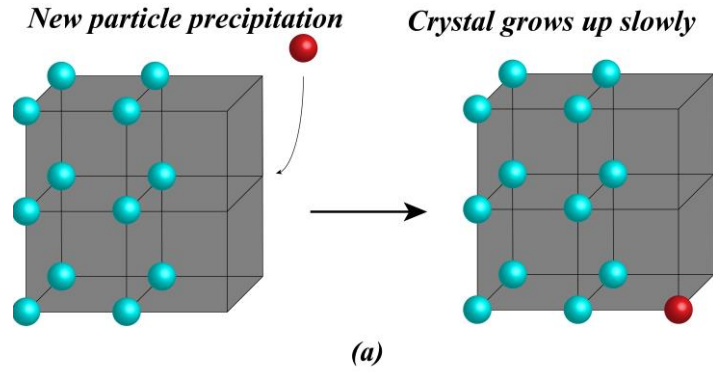
In this formula, the numerator is used to represent the coupling strength of  $PCU_i$  to all other  $PCU$ , and the denominator is used to represent the coupling strength of  $PCU_i$  subject to all other  $PCU$ . When the constraint from  $PCU_i$  to other  $PCU$  is stronger and  $PCU_i$  is less restricted, we see that the atom diameter of it is bigger. For atoms with larger diameters, we think they are easier to crystallize. Here the probability of crystallization is defined to be

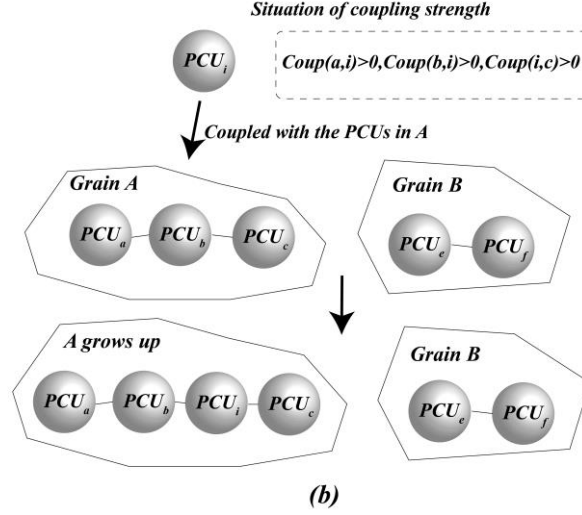
$$p_i = \frac{D_i}{\sum_{i=1}^n D_i} \quad (4)$$

According to the precipitation probability of each PCU atom, a particle is randomly selected to nucleate in this step. The definition of atom diameter and precipitation probability ensures the order in the crystal precipitation process, which helps reduce the quantity of complex crystal fusion steps and improves the efficiency of the subsequent sequence optimization process.

## (2) Crystal growth

This step simulates the process of atom precipitation and finding niches on the surface of the existing crystal, making the crystal continue to grow, as shown in Figure 4 (a).





**Fig. 4.** Comparison of crystal growth process. (a) Process of crystal growth in the nature, (b) Process of the PCU-formed crystal growth in configuration design.

Same as the crystal nucleation process, firstly the precipitation probability of each PCU is calculated

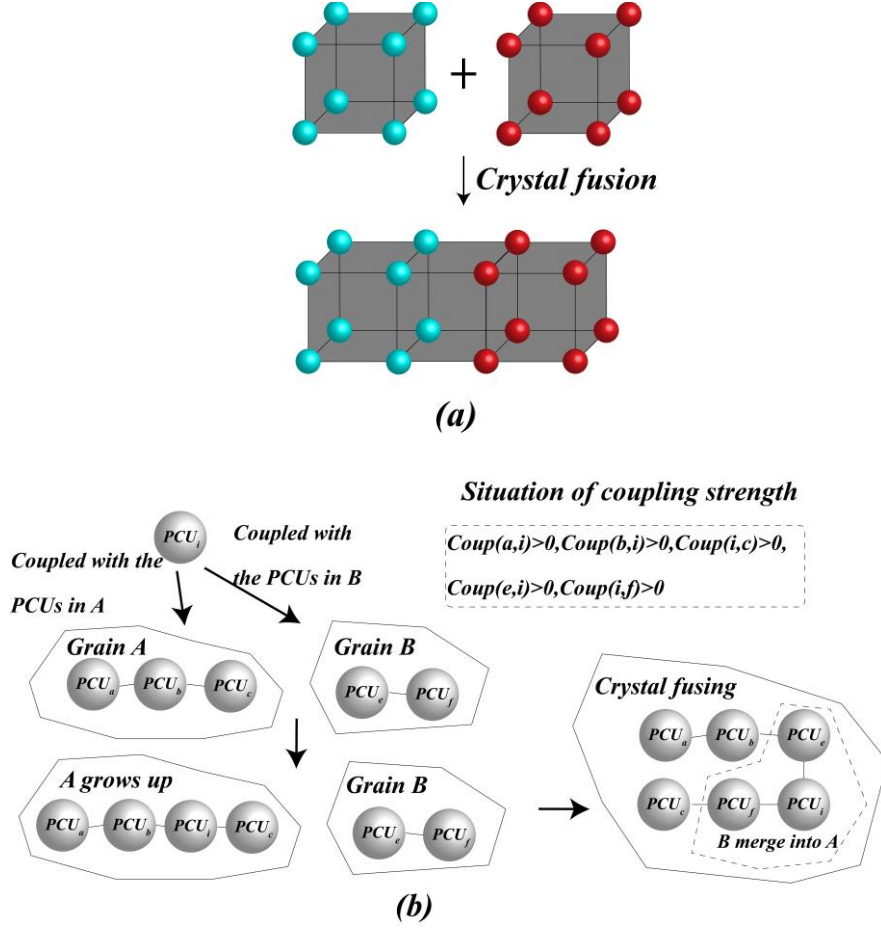
$$p_i = \frac{D_i}{\sum_{i=1}^{n-k} D_i} \quad (5)$$

Where  $k$  is the total quantity of PCUs that have already been precipitated.

According to the probability, in this step we randomly select a PCU that has not yet precipitated, and integrate it into the corresponding position of the crystal to make the crystal grow up. As an example, the detailed implementation steps are revealed in Figure 4 (b). From the figure, it is known that the currently generated grains are  $A = \{PCU_a, PCU_b, PCU_c\}$  and  $B = \{PCU_d, PCU_e\}$ . Now, selecting  $PCU_i$  from the pool of PCU and judging the coupling condition between  $PCU_i$  and the PCUs inside  $A$ ,  $B$ , we can get that  $coup(a,i) > 0$ ,  $coup(b,i) > 0$ ,  $coup(i,c) > 0$ . Therefore, there exists constraints between  $PCU_i$  and  $A$  grain, so let  $PCU_i$  merge into the grain  $A$ , resulting in the growth of  $A$ . According to the value of coupling strength,  $PCU_a$ ,  $PCU_b$  have constraints on  $PCU_i$ , and  $PCU_i$  has constraint on  $PCU_c$ . Thus, the order of PCU in grains  $A$  is adjusted so that the PCU instantiation order is modified as:  $PCU_a \rightarrow PCU_b \rightarrow PCU_i \rightarrow PCU_c$ .

### (3) Crystal fusion

This step simulates the process in which individual crystal grains scattered everywhere touch and fuse to form large grains during the crystal crystallization, as shown in Figure 5 (a).



**Fig. 5.** Comparison of crystal fusion process. (a) Process of crystal fusion in the nature, (b) Process of the PCU-formed crystal fusion in configuration design.

The same as the previous steps, a PCU is selected in terms of the precipitation probability, as is demonstrated in Figure 5 (b). Then we judge the relationship between the selected PCU and each grain, knowing that  $coup(e,i) > 0, coup(i,f) > 0$ , so it can be seen that  $PCU_i$  is coupled with A、B at the same time. Confronting with this situation, the fusion operation is needed. Therefore, the attraction force between the PCU and grains is defined to be

$$F = \sum_j coup(i, j) + \sum_j coup(j, i) \quad (6)$$

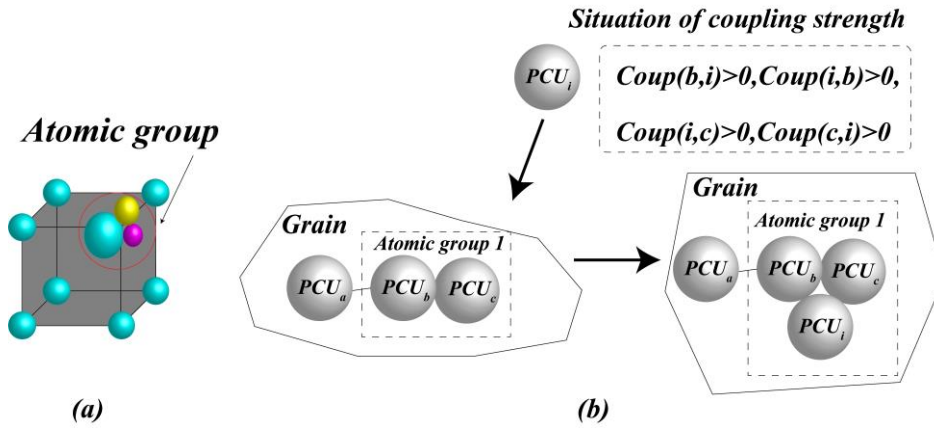
Where  $j$  represents each  $PCU$  in the grain.

The force of attraction between the PCU and each grain is calculated. After that, the grains are arranged in descending order in terms of attraction, and the PCU is integrated into the grain with the greatest attraction. Then, the order of the particles in the A grain is adjusted by constraint relationship. The remaining grains are integrated into the grown grains in order of attraction. After the fusion is completed, a new larger grain is obtained, and the order of arrangement inside the grain is  $PCU_a \rightarrow PCU_b \rightarrow PCU_e \rightarrow PCU_i \rightarrow PCU_f \rightarrow PCU_c$ .

(4) Atomic group formation

As shown in Figure 6 (a), sometimes it is not a single atom that participates in the crystallization

in the nature, but may be a group of atoms, such as  $SO_4^{2-}$ ,  $NH_4^+$ . They are called atomic groups in chemistry. For the situation shown in Figure 2 (c) in Section 3, when the directed graph of the coupling relationship contains strongly connected components, it is difficult to determine the instantiation order of each PCU based solely on constraints. For this type of problem, the solution is to identify all strongly connected components in the directed graph, the PCUs in the strongly connected components are considered as a whole and consolidated into an atomic group. Figure 2 (c) contains two strongly connected components. It can be identified as two atomic groups. Then, the constraint relationship of atomic group-atomic group and atomic group-atom are utilized to determine the arrangement position of the atomic group in the grain. After the crystallization is completed, the method proposed later is used to decouple the complex coupled PCU inside the atomic group and determine the instantiation sequence.



**Fig. 6.** Comparison of atomic group. (a) Atomic group in the nature, (b) Process of the Atomic group formation in configuration design.

The identification method of strongly connected components in the directed graph is shown in Figure 6 (b). When  $PCU_i$  and  $PCU_j$  are in the same strongly connected component, there must be  $coup(i, j) > 0$  and  $coup(j, i) > 0$ . It can be seen from the figure that the coupling strength between  $PCU_i$  and the two  $PCUs$  in the atomic group is  $coup(b, i) > 0$ ,  $coup(i, b) > 0$ ,  $coup(c, i) > 0$ ,  $coup(i, c) > 0$ . Therefore,  $PCU_i$ ,  $PCU_b$  and  $PCU_c$  are merged into a same atomic group. In fact, according to the definition of strongly connected components,  $PCU_i$  is only required to satisfy  $coup(i, k) > 0$  and  $coup(k, i) > 0$  with any  $PCU_k$  in the atomic group to reach the conclusion that  $PCU_i$  is in the current strongly connected component.

##### (5) Atomic group decoupling

There is bi-directional coupling relationship between PCUs in the atomic group. For  $PCU_i$  and  $PCU_j$ , if the instantiation of  $PCU_i$  needs output information from  $PCU_j$ , and the instantiation of  $PCU_j$  needs output information from  $PCU_i$ , the configuration solving process will be complex. To complete the instantiation of the two, we first need to make assumptions about

the information of  $PCU_i$  and pass it to  $PCU_j$  in order to completing the instantiation of  $PCU_j$ . Then, the information of  $PCU_j$  has to be passed to  $PCU_i$  to complete the instantiation of  $PCU_i$ . If the information of  $PCU_i$  after instantiation is inconsistent with the hypothetical information, a new hypothesis must be made and the two must be instantiated again. At this time, there exists iteration in the configuration solving process. In order to reduce iterations and improve the efficiency, it is necessary to reasonably plan instantiation sequence of the PCUs in atomic groups, which is the so-called decoupling process. According to Smith and Eppinger [20], for such coupled tasks, the sequence planning should satisfy the following two propositions

**Proposition 1.** When other conditions are the same, the task with larger output and smaller input should be executed before the task with smaller output and larger input.

**Proposition 2.** When other conditions are the same, tasks with a shorter execution time should be executed before tasks with a longer execution time.

Based on the above propositions, Zhiliang et al.[25] propose an instantiation sequence optimization method based on time-coupling degree, which is used here to decouple atomic groups.

Suppose the PCU set in the atomic group is  $\{PCU_1, PCU_2, \dots, PCU_k\}$ , and the time required for each PCU to complete the instantiation is  $\{T_1, T_2, \dots, T_k\}$ . When  $PCU_j$  in the atomic group is instantiated after other  $PCUs$ , it may prompt other  $PCUs$  to re-instantiate. The sum of the expected time required for re-instantiation is

$$R_{j1} = \sum_{i=1, i \neq j}^k coup(j, i)T_i \quad (7)$$

In the same way, when  $PCU_j$  is instantiated before other  $PCUs$ , other  $PCUs$  may prompt  $PCU_j$  to be instantiated again, and the total expected time required is

$$R_{j2} = \sum_{i=1, i \neq j}^k coup(i, j)T_j \quad (8)$$

The larger the value of  $R_{j1}$  is, the earlier the  $PCU_j$  should be instantiated. The larger the value of  $R_{j2}$  is, the later the  $PCU_j$  should be instantiated. We use  $G_j$  to measure the instantiation priority of  $PCU_j$

$$G_j = \frac{\sum_{i=1, i \neq j}^k coup(j, i)T_i}{\sum_{i=1, i \neq j}^k coup(i, j)T_j} = \frac{R_{j1}}{R_{j2}} \quad (9)$$

The larger the value of  $G_j$  is, the earlier the  $PCU_j$  should be instantiated.

When the PCU is instantiated according to the configuration rules, the execution time of the configuration solving process is positively related to the number of configuration rules encapsulated in the PCU:



$$T \propto n \quad (10)$$

Where  $n$  is the total number of configuration rules.

We use  $n$  instead of time  $T$  and rewrite  $G_j$  as

$$G_j' = \frac{\sum_{i=1, i \neq j}^k \text{coup}(j, i) n_i}{\sum_{i=1, i \neq j}^k \text{coup}(i, j) n_j} \quad (11)$$

Taking account of  $G_j'$ , the instantiation sequence of the PCU instantiation task is determined as follows

**Step 1.** Let  $Q$  be the PCU instantiation sequence, and the length of  $Q$  is  $k=0$ .

**Step 2.** For each  $PCU_j$  in the atomic group, calculate  $G_j'$ .

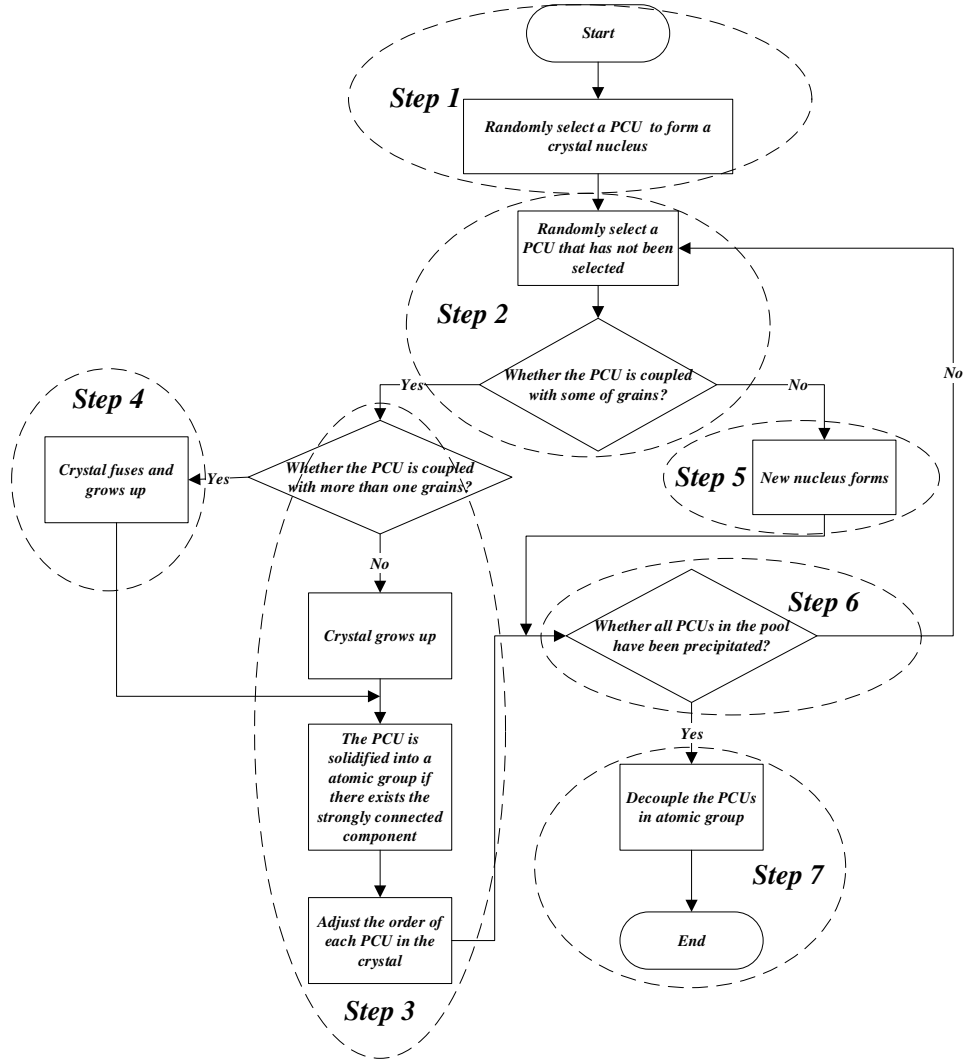
**Step 3.** Select the  $PCU_m$  whose  $G_m' = \max\{G_j'\} (j=1, 2, \dots, k-l)$ .  $l$  is the number of PCUs that have already been sorted.

**Step 4.** Place  $PCU_m$  at the end of the sequence  $Q$  and delete  $PCU_m$  from the atomic group.

**Step 5.** Determine whether all PCUs in the atomic group have been sorted. If the sorting has been finished, end the loop. If the sorting has not been finished, jump to step 2.

### 5.2. The process of product configuration solving process planning

The arrangement of atoms and atomic groups in the crystal always has a remarkable regularity, so is the order of each PCU in the PCT after arrangement. By imitating the CCP, we can get the regular arrangement of each PCU, which is the improved sequence of product configuration solving process. Figure 7 gives the flowchart of our method. Each PCU is regarded as an atom in the pool. Through the processes of crystal nucleation, crystal growth, crystal fusion, all PCUs form a regular microstructure to obtain a large crystal.



**Fig. 7.** Flowchart of the heuristic method.

For any PCT, the detailed implementation steps of the algorithm are as follows:

**Step 1.** According to the precipitation probability, select a PCU from the PCU pool to form a crystal nucleus as the starting point for crystal growth.

**Step 2.** Choose a PCU from the current PCU pool based on the probability and judge its relationship with each grain. If it is coupled with one of the grains, skip to step 3; if it is coupled with more than one grains, skip to step 4; if it is uncoupled with any grain, skip to step 5.

**Step 3.** Merge this PCU into the grain and judge whether the PCU forms a strongly connected component with other PCUs inside the grain. If there exists the strongly connected component, the PCU is solidified into an atomic group, treat the atomic group as a whole and the order of arrangement is adjusted according to the constraint relationship. If there exists no strongly connected component, adjust the order directly by evaluating the relationship between PCUs.

**Step 4.** Calculate the attractive force between the PCU and each grain, add the PCU to the grain with the biggest attractive force and perform step 3, and then merge the rest of the grains into the grown grain in order by the attractive force to complete the crystal fusion step.

**Step 5.** If the PCU is uncoupled with the existing grains, let this PCU nucleate as a new starting

point for crystal crystallization.

**Step 6.** Judge whether all PCUs in the pool have been precipitated. If there are still PCUs remaining in the pool, skip to step 2; else skip to step 7.

**Step 7.** Take out the obtained atomic groups, use the tearing algorithm informed in Section 5.1 to decouple the PCU, and determine the arrangement order of the PCUs in the atomic group. The configuration solving process planning is completed.

After the crystallization is completed, each PCU is combined to form one or more grains, and all grains form a big crystal. The crystal grains are in a parallel relationship and can be executed synchronously during configuration. Each PCU in the grain is instantiated in sequence according to their arrangement order.

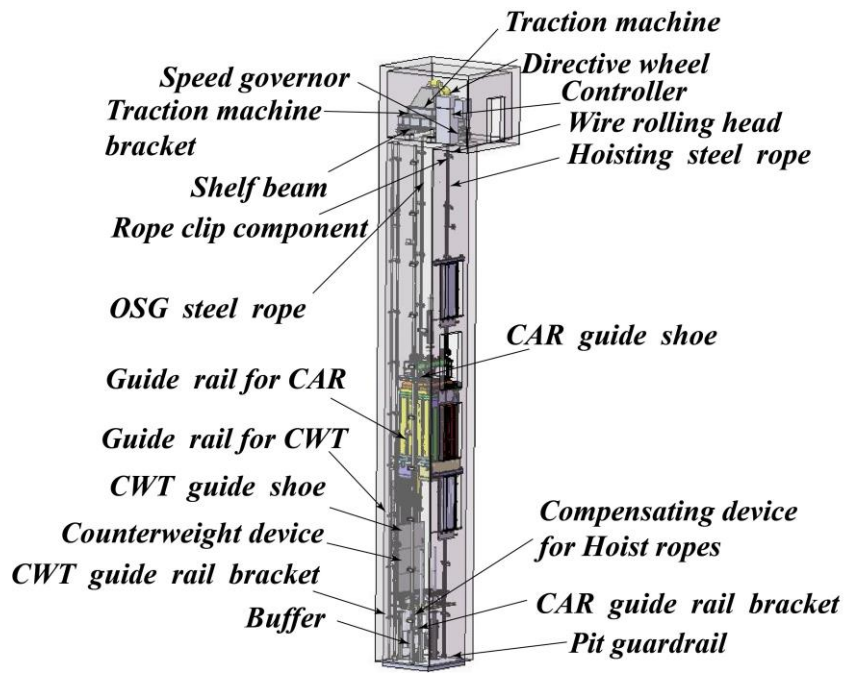
## 6. Verification and results analysis

### 6.1. An industrial illustrative example of configuration design

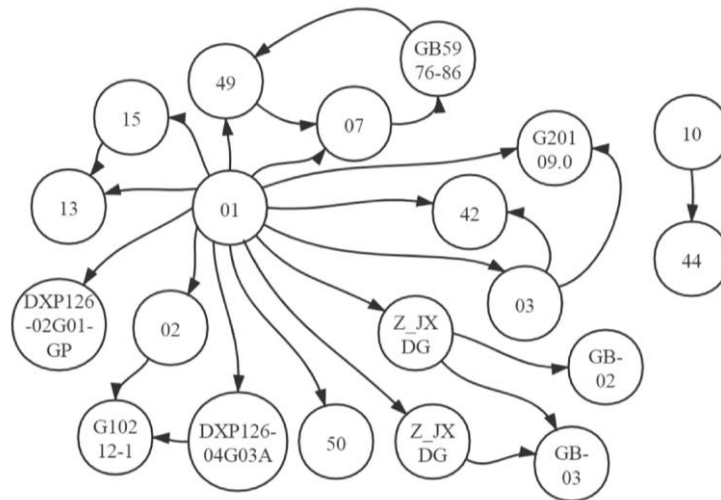
Mechanical product configuration design is especially suitable for the industry where customer demands change continuously and the product can easily be modularized. The design of elevators is a high-variety, low-volume market in which competition is very high [26]. Elevator products are typical order-type products, with significant functions and structural levels. The product configuration design of the elevator can be realized through modularization. Customers provide their requirements, then enterprise combine different PCUs to generate PCT and instantiate each PCU in PCT. To increase the efficiency and accuracy of configuration design, it is necessary to assign a reasonable instantiation order of each PCU. This paper takes the GB10 elevator produced by Canny Elevator as an example, selects some of the coupled PCUs, and optimizes the instantiation sequence. The involved PCUs are shown in Table 1, the elevator model is shown in Figure 8 (a), and the directed graph of the constraint relationship between the PCUs is shown in Figure 8 (b).

**Table 1 Coupled PCUs**

PCU ID	PCU name	PCU ID	PCU name
01	Traction machine	GB-02	Car guide rail bracket
49	Buffer	Z-DZDG	Guide rail for CWT
GB5976-86	Wire rolling head	DXP126-04G03A	CAR guide rail
07	Speed governor	02	Traction machine bracket
42	Counterweight device	G10212-1	Shelf beam
G20109.0	Controller	DXP126-02G01-GP	CWT guide shoe
03	Directive wheel	13	Hoisting steel rope
50	Pit guardrail	15	Rope clip component
Z-JXDG	Guide rail for CAR	10	OSG steel rope
GB-03	CWT guide rail bracket	44	Compensating device for Hoist ropes



(a)



(b)

**Fig. 8.** PCU involved in the configuration process. (a) Three-dimensional model of elevator, (b) Directed graph of PCUs.

(1) Configuration unit constraint analysis and coupling strength calculation

Given the configuration rules encapsulated in the PCU, the method in section 4.3 is used to calculate the weight of each PCU's attribute parameters and the frequent item of parameter combination. Take the traction machine as an example, the calculation results are shown in Table 2.

In the table, the single parameter represents one of the attribute parameters of the traction machine that restricts other PCUs, and the parameter combination is the frequent item mined by the FP-growth algorithm.

**Table 2** Weights of traction machine's attribute parameters and their combination frequent

items

PCU ID	PCU name	Attribute id	Attribute name	Frequency	Weight
		standard	Standard	34	0.0251
		BPQLX	Inverter type	36	0.0265
01	Traction machine	YYBPD	Traction ratio	35	0.0258
		N1	The number of traction wheel rope-groove	78	0.0575
...	...	...	...	...	...
...	...	...	...	...	...
		V、BPQLX	Nominal velocity, Inverter type	30	0.0221
		N1、YYBPD	The number of traction wheel rope-groove, Traction ratio	29	0.0214
		V、YYJCJ	Nominal velocity, traction machine manufacturers	50	0.0369

Knowing the weights and configuration rules, the coupling strength between PCUs can be calculated. Take the rope clip component as an example, the configuration rule of it is

```

IF
(N1 = 3 OR N1 = 5)AND(YYBPD = 1)AND(DZWZ = rear)
THEN
selected _id = T01400601-001

```

It can be seen from the <prerequisite clause> part of the configuration rule that the instantiation of the rope clamp component requires the parameters of *N1* and *YYBPD* from traction machine. From the previous calculation we can get that the weight of *N1* is 0.0575 and the weight of *YYBPD* is 0.0258. At the same time, there exists frequent itemsets {*N1*, *YYBPD*}, the weight of it is 0.0214. Therefore, the coupling strength of the traction machine to the rope clip component is 0.1047. The coupling strength is calculated in this way for all other PCUs, and the calculation results are shown in Table 3.

**Table 3 Coupling strength between PCUs**

Restrict PCU ID	Restrict PCU name	Restricted PCU ID	Restricted PCU name	Coupling parameters	Coupling Strength
01	Traction machine	15	Rope clip component	N1、YYBPD	0.1047

01	Traction machine	50	Pit guardrail	Q、standard、DZKLX、V	0.3842
01	Traction machine	49	Buffer	V、Q	0.3370
01	Traction machine	GB5976-86	Wire rolling head	V、Q	0.3370
...	...	...	...	...	...
...	...	...	...	...	...
07	Speed governor	GB5976-86	Wire rolling head	KXSQLX	0.0808
GB5976-86	Wire rolling head	49	Buffer	XSQSZJ	0.1765
02	Traction machine bracket	G10212-1	Shelf beam	H2	0.8792

## (2) Instantiation sequence arrangement

Knowing the coupling strength between PCUs, the method proposed in Section 5 is used to optimize the instantiation sequence of the PCUs. The crystal growth process is as follows.

$$\begin{aligned}
 & [] \xrightarrow{a} [Z\_DZDG] \xrightarrow{a} [Z\_DZDG],[10] \xrightarrow{b} [01,Z\_DZDG],[10] \xrightarrow{a} \\
 & [01,Z\_DZDG],[10],[50] \xrightarrow{c} [01,49,50,Z\_DZDG],[10] \xrightarrow{b} [01,02,07,49,50,Z\_DZDG],[10] \\
 & \xrightarrow{d} [01,02,[GB5976-86,07,49],50,Z\_DZDG],[10] \rightarrow \dots \rightarrow [10,44], [01,15,13,DXP126-04G03A, \\
 & 03,42, DXP126-02G01-GP,Z\_JXDG, GB-02,G20109.0,02, G10212-1,[GB5976-86,07,49],50,Z\_DZDG,GB-03] \\
 & \xrightarrow{e} [49,07,GB5976-86]
 \end{aligned}$$

The PCU precipitated each time is randomly selected according to the calculated precipitation probability.

Symbol *a* represents the nucleation process. For example, when PCU 10 is selected, we find that the existing grain [Z\_DZDG] has no constraint with 10, thus PCU 10 forms a new grain.

Symbol *b* represents the crystal growth process. For example, when PCU 01 is selected, we find that  $coup(01,Z\_DZDG)=0.4319$ . Thus, we can judge that PCU 01 has constraints with [Z\_DZDG]. PCU 01 is integrated into [Z\_DZDG] and is placed in front of PCU Z\_DZDG according to the direction of the constraint.

Symbol *c* represents the crystal process. For example, when PCU 49 is selected, according to coupling strength, we judge it has constraints with both [01, Z\_DZDG] and [50]. The attractive force is calculated and we find that [50] is more attractive to PCU 49. Thus, PCU 49 is firstly merge into [50] and get [49, 50]. Then, [49, 50] and [01, Z\_DZDG] are fused and [01, 49, 50, Z\_DZDG] is gotten.

Symbol *d* represents the atomic group formation process. Bidirectional constraints among GB5976-86, 07 and 49 have been detected and atomic group [GB-5976-86, 07, 49] is gotten.

Symbol *e* represents the decoupling process for atomic group [GB-5976-86, 07, 49]. Using the

method in Section 5.1(5), we can get the sequence of these PCUs to be [49, 07, GB5976-86].

Two grains are gotten after crystallization. The relationship between the two grains is parallel and they can be solved at the same time. The relationship between PCUs in the grain is sequential and they have to be instantiated in sequence.

## 6.2. Comparison and discussion

In order to verify the correctness and effectiveness of the PCU instantiation sequence in configuration solving process gotten by our method, some other instantiation sequence is used to instantiate the PCUs in PCT. These sequences include: three common ways to traverse the PCT tree and six other random sequences. The number of mismatched PCUs (PCUs that fail to match the instance) is compared after the configuration solving process is completed. The result are shown in Table 5. As we can see from the table, only 2 PCUs mismatch using the sequence we got to instantiate PCUs, which is obviously better than other instantiation sequences.

**Table 4 Comparison of the configuration solving results using different instantiation sequence**

PCU instantiation sequence	Number of mismatched PCUs	PCU instantiation sequence	Number of mismatched PCUs
Sequence gotten by our method	2	Random sequence 2	5
The preorder traversal	5	Random sequence 3	13
The inorder traversal	12	Random sequence 4	8
The postorder traversal	15	Random sequence 5	16
Random sequence 1	6	Random sequence 6	11

For the three PCUs in the atomic group, 49, GB5976-86 and 07, the instantiation order has  $3! = 6$  combinations. In order to verify that the optimal instantiation order obtained by the above method with the target of minimum time is [49,07, GB5976-86], each PCU is instantiated 20 times according to all 6 order combinations, the time it takes to complete the instantiation is recorded. Then, the average time-consuming of 20 simulation results is calculated, and the results are shown in Table 5.

**Table 5 Average time-consuming of configuration design with each sequence**

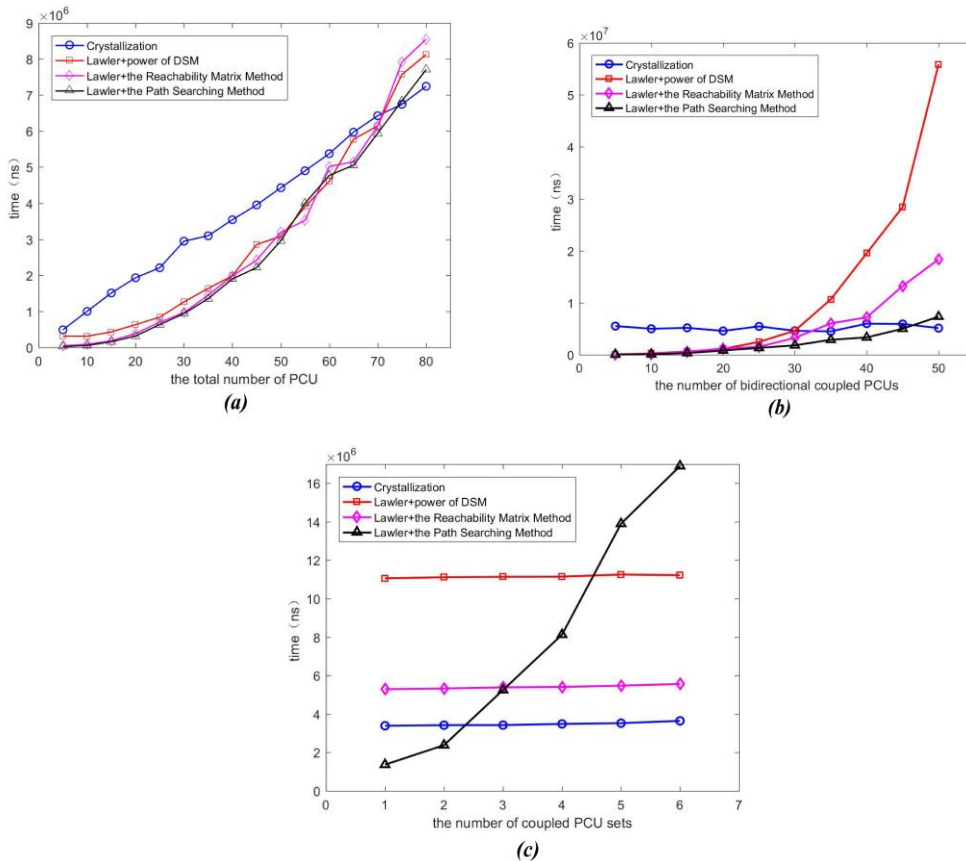
Instantiation sequence	average time-consuming (ms)	Instantiation sequence	average time-consuming (ms)
[49,07, GB5976-86]	7479.6	[07,49,GB5976-86]	8615.2
[49,GB5976-86,07]	7625.7	[GB5976-86,07,49]	9238.2
[GB5976-86,49,07]	8413.1	[07,GB5976-86,49]	9374.1

As shown in the table, the instantiation sequence [49, 07, GB5976-86] does take the minimum time. Instantiating in the order of decoupling can effectively reduce the number of iterations of the configuration solving process, reduce configuration time, and improve configuration efficiency.

The modeling method using DSM is the most frequently utilized activities management

framework. This management framework usually includes processes such as sorting, partition, and tearing. The object of the sorting algorithm is to sort tasks that do not contain a bidirectional coupling relationship, and the most widely used algorithm is proposed by Lawler [27]. The goal of the decomposition algorithm is to find the smallest block matrix, that is, to identify the coupled task set. The frequently-used methods include the power of DSM [28,29], Reachability Matrix [30,31], path searching [32,33] and some heuristic algorithms such as the Simulated Annealing [34], Genetic Algorithm [35,36]. The DSM sorting and partition corresponds to the process of crystal nucleation, growth, fusion and atomic group identification in this paper. Without considering the subsequent atomic group decoupling process, our method and the above methods can get the almost same sequence. Based on that, the method in this paper is compared with other algorithms in terms of the running time.

The efficiency of the method is mainly affected by three factors: the number of PCUs included in the PCT, the number of PCUs with bidirectional coupling, and the number of coupled PCU sets. In order to compare the method in this paper with other algorithms, each factor is changed individually and different algorithms are used to plan the PCU instantiation sequence. The running time of each algorithm is recorded and the results are shown in Figure 9.



**Fig. 9.** Running time of each method with different conditions. (a) the total number of PCUs alters, (b) the number of PCUs with bi-directional coupling alters, (c) the number of coupled PCU sets alters.

Figure 9 (a) corresponds to the condition where the number of bi-directionally coupled PCUs



is 5, the number of coupled PCU sets is 2, and the total number of PCUs changes. It can be seen from the figure that, when the total number of PCUs is less than 60, the computational efficiency of power of DSM method, Reachability Matrix method and path search method is higher, but when the number of PCUs grows to 60-70, the algorithm proposed in this paper gradually exceeds the other algorithms. That is to say, our method has greater advantages when the number of PCUs is large. This is because the crystallization operations of our method are more time-consuming when the number of PCUs is small, and the flexibility of the method in this paper begins to show its advantages when the number of PCUs becomes larger.

Figure 9 (b) corresponds to the condition where the total number of PCUs is 50, the number of coupled PCU sets is 2, and the number of PCUs with bidirectional coupling changes. This figure reveals that when the number of bi-directionally coupled PCUs increases, the running time of the traditional algorithms greatly increase, while the running time of the algorithm in this paper is almost unaffected. It can also be found that, compared with the traditional DSM sorting and partition algorithm, the running time of our method is greatly affected by the total number of PCUs, but less affected by the complexity of the coupled PCU set. When the number of PCUs is large and the coupling relationship is more complicated, our method has obvious advantages over the above algorithms.

Figure 9 (c) corresponds to the condition where the total number of PCUs is 50, the number of bi-directionally coupled PCUs is 35, and the number of coupled PCU sets changes. When the number of coupled PCU sets increases, the efficiency of the algorithm in this paper, the power of DSM method, and the reachability matrix method does not change much, while the efficiency of the path search algorithm is greatly reduced. The reason for this phenomenon is that the other three algorithms perform one-step operations to identify all coupled PCU sets. As for the path search algorithm, when there are more than one coupled PCU set, it requires to go through more search steps to identify all sets.

From the above experimental conclusions, it can be known that power of DSM method, reachability matrix method, and path search method are more efficient for simple DSM matrix, but are not ideal for complex DSM. The path search algorithm starts from a random PCU and searches for a coupled PCU set along the path according to the correlation between the PCUs. When the number of PCUs increases, the randomness of the path search will be greatly increased, which can lead to the increase of search time. The power of DSM method uses continuous matrix multiplication. When the matrix dimension increases, the calculation time will increase greatly. The reachability matrix method uses the method of seeking strong connectivity in graph theory to find the coupled PCU sets. Its time complexity is  $O(n^2)$ . All the three algorithms operate on the DSM matrix as a whole. It's also worth mentioning that although this paper did not conduct corresponding experiments, when using the power of DSM method and the reachability matrix method to arrange the coupled PCU set, if there are multiple coupled sets of the same length in the DSM or when a design activity is multiple, there exists the problem that the result is not the smallest coupled PCU

set. The algorithm in this paper does not have the above problems.

Genetic algorithm is a type of heuristic algorithm, which searches for the optimal solution of the problem by imitating the evolutionary reproduction process of biology, and has been widely used to arrange design activities in recent years. When genetic algorithm is applied to the problem discussed in this paper, it can get similar results. However, when applying genetic algorithms, the choice of parameters has a significant impact on the efficiency of the algorithm. In this paper we further compare our method with the genetic algorithm by calculating the running time of each algorithm, in which the mutation probability of GA is fixed to be 0.001 and the population size of GA is changed. The result has been displayed in Table6.

**Table 6 Efficiency comparison between the genetic algorithm and the proposed algorithm**

Population size	PCU quantity	The generation of population reproduction	Time-consuming ( <i>ns</i> )	Time-consuming of our method( <i>ns</i> )
20	5	2	$1.02 \times 10^5$	$4.98 \times 10^5$
	10	56	$7.15 \times 10^6$	$9.7 \times 10^5$
	15	76	$1.36 \times 10^7$	$1.45 \times 10^6$
	20	117	$2.49 \times 10^7$	$1.92 \times 10^6$
60	5	1	$3.47 \times 10^5$	$4.98 \times 10^5$
	10	19	$4.25 \times 10^6$	$9.7 \times 10^5$
	15	33	$9.31 \times 10^6$	$1.45 \times 10^6$
	20	68	$2.08 \times 10^7$	$1.92 \times 10^6$
100	5	0	$4.21 \times 10^5$	$4.98 \times 10^5$
	10	16	$6.72 \times 10^6$	$9.7 \times 10^5$
	15	28	$1.05 \times 10^7$	$1.45 \times 10^6$
	20	36	$1.99 \times 10^7$	$1.92 \times 10^6$

Obviously, when the genetic algorithm is used for design activity planning, the choice of parameters has a great impact on its efficiency. When the number of PCUs is small, a smaller population size should be selected. At this time, the efficiency of the genetic algorithm is higher than the method proposed in this paper; when the number of PCUs is large, a larger population size should be selected, but the genetic algorithm's efficiency is lower than our method. For optimization problems with complex combinations, genetic algorithms have a large search space and long search time, and are prone to premature convergence. Compared with genetic algorithm, which is also a heuristic algorithm, the method in this paper does not have the problem of parameter selection, and it is more stable for complex problems.

## 7. Conclusion

This paper summarizes and abstracts the product configuration methods of machinery manufacturers. Aiming at the configuration of complex mechanical products, concepts such as PCU and PCT are defined, and a product configuration design model is established to meet the

requirements of mass customization. For the problem of PCU instantiation sequence that appears during the configuration solving process, a heuristic configuration solving process planning method for product configuration design by imitating the CCP is proposed.

The biggest contribution of this paper is that when machinery manufacturers like Canny Elevator design configurators, it provides them with a method to make the configuration solving process more reasonable. Compared with traditional methods, the results of our coupling calculation method has stronger objectivity. Our heuristic method can effectively avoid the cumbersome characteristics of traditional DSM related algorithms and has great advantages for complex mechanical products with various component types. In addition to configuration design, this method can also be used to plan and coordinate other activities such as production activities. Despite the benefits outlined above, our research still has its own limitations. As for the calculation of coupling strength between PCUs, although the subjective factors are avoided, the amount of calculation will increase obviously when the mechanical product has quantities of configuration rules. This might be the direction for future improvement.

### **Acknowledgements**

This work has been funded by the National Key R&D Program of China (2018YFB1700700), the National Natural Science Foundation of China (51905476), the Jiangsu Province Science and Technology Achievement Transforming Fund Project (BA2018083).

### **Declaration of conflicting interests**

Authors declare that they have no conflict of interest.

### **Ethical Approval**

No applicable

### **Consent to Participate**

No applicable

### **Consent to Publish**

Authors consent to publish this article

### **References**

1. Wei W, Fan W, Li Z (2014) Multi-objective optimization and evaluation method of modular product configuration design scheme. *Int J Adv Manuf Technol* 75 (9):1527-1536
2. Kott A, Agin G, Fawcett D (1992) Configuration Tree Solver - a Technology for Automated Design and Configuration. *J Mech Design* 114 (1):187-195
3. Yang D, Dong M, Miao R (2008) Development of a product configuration system with an ontology-based approach. *Comput Aided Design* 40 (8):863-878
4. Steward DV (1981) The Design Structure-System - a Method for Managing the Design of Complex-Systems. *Ieee T Eng Manage* 28 (3):71-74

5. Mcdermott J (1982) R1 - a Rule-Based Configurer of Computer-Systems. *Artif Intell* 19 (1):39-88
6. Soloway E, Bachant J, Jensen K (1987) Assessing the maintainability of XCON-in-RIME: coping with the problems of a VERY large rule-base. In *Proceedings of the sixth National conference on Artificial intelligence - Volume 2*, Seattle, Washington, pp 824–829.
7. Felfernig A, Hotz L, Bagley C, Tiuhonen J (2014) *Knowledge-based Configuration: From Research to Business Cases*. Morgan Kaufmann Publishers Inc, San Francisco
8. Mittal S, Frayman F (1989) Towards a generic model of configuraton tasks. In the *Proceedings of the 11th international joint conference on Artificial intelligence - Volume 2*, Detroit, Michigan, pp 1395-1401.
9. Mailharro D (1998) A classification and constraint-based framework for configuration. *Ai Edam* 12 (4):383-397
10. Amilhastre J, Fargier H, Marquis P (2002) Consistency restoration and explanations in dynamic CSPs-Application to configuration. *Artif Intell* 135 (1-2):199-234
11. Wang P, Gong Y, Xie H, Liu Y (2016) SOMEDGRA: A case retrieval method for machine tool product configuration design. *J Mech Sci Technol* 30 (7):3283-3293
12. Zeng FS, Jin Y (2007) Study on product configuration based on product model. *The International Int J Adv Manuf Technol* 33 (7):766-771
13. Lee J, Han S (2010) Knowledge-based configuration design of a train bogie. *J Mech Sci Technol* 24 (12):2503-2510
14. Dong M, Yang D, Su LY (2011) Ontology-based service product configuration system modeling and development. *Expert Syst Appl* 38 (9):11770-11786
15. Yassine A, Falkenburg D, Chelst K (1999) Engineering design management: an information structure approach. *Int J Prod Res* 37 (13):2957-2975
16. Srour IM, Abdul-Malak MAU, Yassine AA, Ramadan M (2013) A methodology for scheduling overlapped design activities based on dependency information. *Automat Constr* 29:1-11
17. Lin J, Qian YJ, Yassine AA, Cui WT (2012) A fuzzy approach for sequencing interrelated activities in a DSM. *Int J Prod Res* 50 (23):7012-7025
18. Spinner M (1989) *Improving project management skills and techniques*. vol Book, Whole. Prentice Hall, Englewood Cliffs, N.J
19. Shishko R. AR, and Cassingham R.C (1995) *NASA Systems Engineering Handbook*. Washington, D.C.: National Aeronautics and Space Administration
20. Smith RP, Eppinger SD (1997) A predictive model of sequential iteration in engineering design. *Manage Sci* 43 (8):1104-1120
21. Huang JQ, Li YP, Chu XN, Chu DX (2015) An integrated top-down design process evaluation approach of complex products and systems based on hierarchical design structure matrix. *Int J Comput Integ M* 28 (10):1015-1029
22. Chen L, Ding ZD, Li S (2005) A formal two-phase method for decomposition of complex design problems. *J Mech Design* 127 (2):184-195
23. Chen L, Macwan A, Li S (2007) Model-based rapid redesign using decomposition patterns. *J Mech Design* 129 (3):283-294
24. Jafari MH, Kosari A, Fakoor M (2019) A new heuristic method for improved structuring of the Work Transformation Matrix (WTM). *Sci Iran* 26 (4):2506-2523
25. Zhiliang W, Yunxia W, Yun L (2011) Primary Exploration of Optimization Theory and Method for Ordering Coupled Tasks. [*Chinese Journal of Mechanical Engineering*] 22 (12):1444-1449(in chinese)
26. Lee DM, Drake PR (2010) A portfolio model for component purchasing strategy and the case study of two South Korean elevator manufacturers. *Int J Prod Res* 48 (22):6651-6682
27. Lawler EL (1976) *Combinatorial optimization : networks and matroids*. Holt, Reinhardt and Winston, New York
28. Warfield JN (1973) Binary Matrices in System Modeling. *Ieee T Syst Man Cyb Smc3* (5):441-449
29. Sharman DM, Yassine AA (2004) Characterizing complex product architectures. *Systems Eng* 7 (1):35-60
30. Gebala D, Eppinger S (1991) Methods for Analyzing Design Procedures. In *ASME design theory and methodology* pp 39–46
31. Kosztyán ZT (2015) Exact Algorithm for Matrix-Based Multilevel Project Planning Problems. In *Innovations and Advances in Computing, Informatics, Systems Sciences, Networking and Engineering*. Springer International Publishing, pp 487-493
32. Rogers J (1990) Knowledge-Based Tool for Decomposing Complex Design Problems. *J Comput Civil Eng*. doi:10.1061/(ASCE)0887-3801(1990)4:4(298)
33. English KW, Bloebaum CL (2008) Visual Dependency Structure Matrix for Multidisciplinary Design Optimization Tradeoff Studies. *J Aeros Comp Inf Com* 5 (9):274-297

34. Abdelsalam HME, Bao HP (2006) A simulation-based optimization framework for product development cycle time reduction. *Ieee T Eng Manage* 53 (1):69-85
35. Altus SS, Kroo IM, Gage PJ (1996) A genetic algorithm for scheduling and decomposition of multidisciplinary design problems. *J Mech Design* 118 (4):486-489
36. Sun L, An J, Yu H (2016) The Application of Design Structure Matrix Optimization Method Based on Genetic Algorithm. In the International Conference on Communications, Information Management and Network Security, Kuala Lumpur, Malaysia, pp 271-273

# Figures

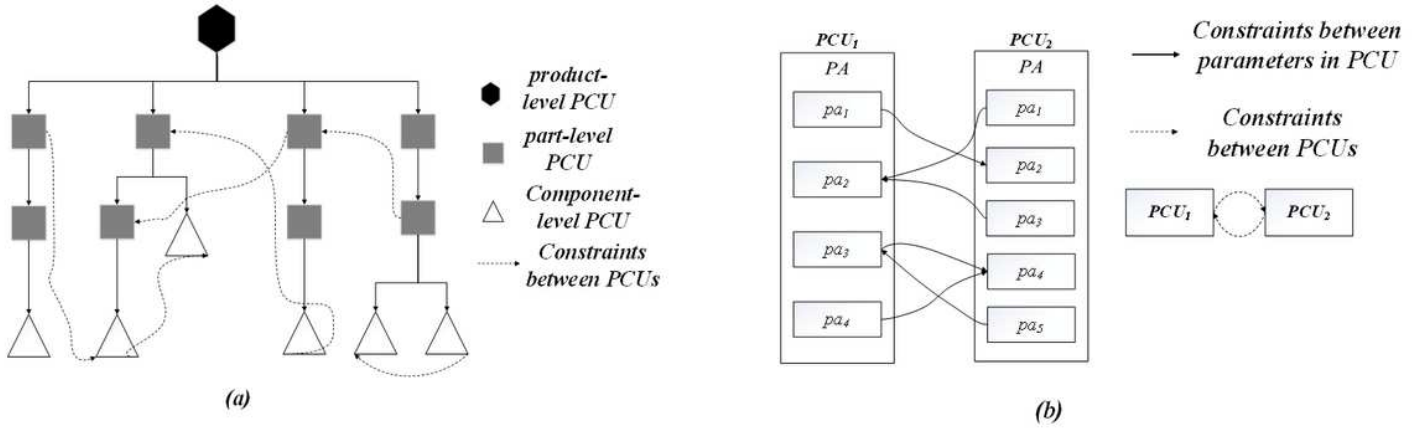


Figure 1

Constraints in PCT. (a) Product configuration template structure tree, (b) Cause of coupling between PCUs.

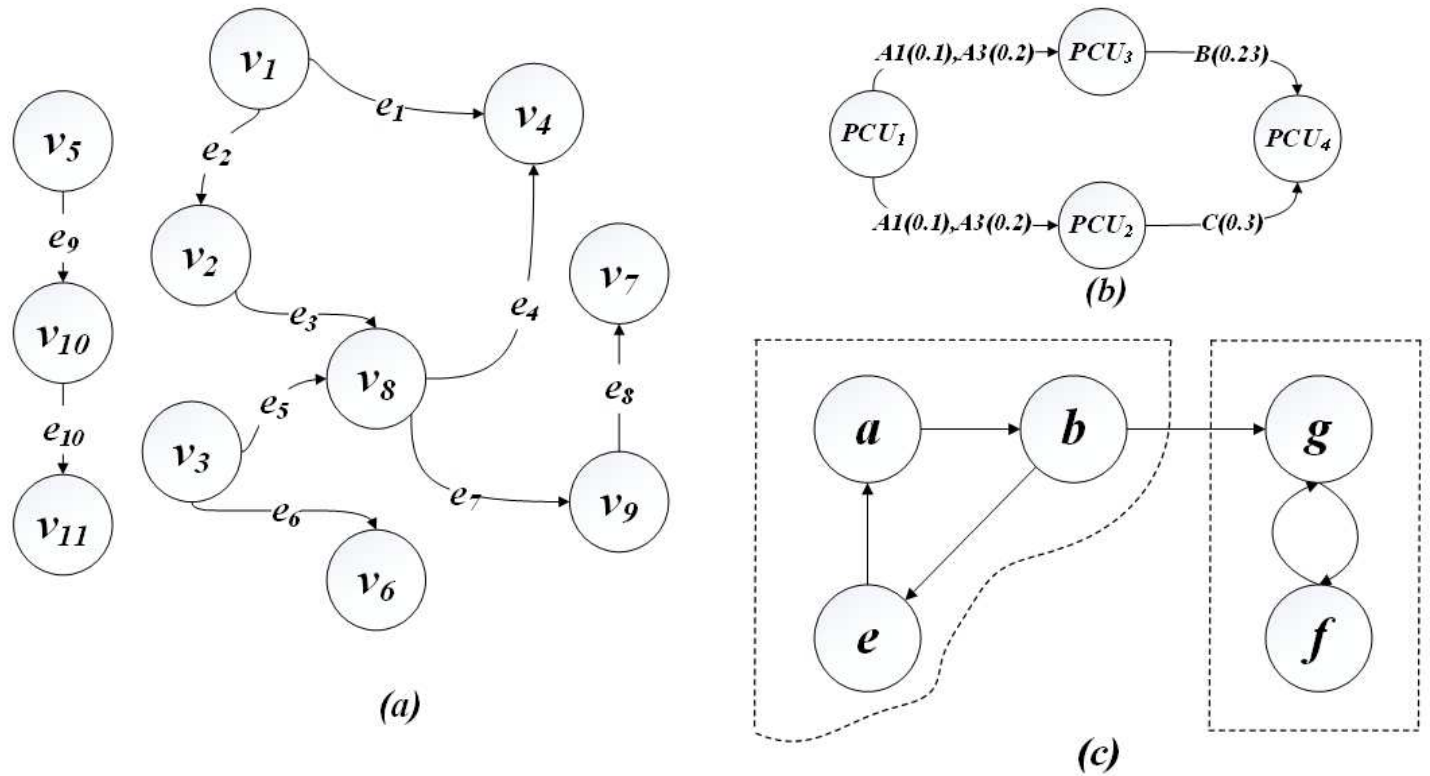


Figure 2

Directed graph of coupling relationship. (a) Common directed graph, (b) Directed graph without strongly connected components, (c) Directed graph with strongly connected components.

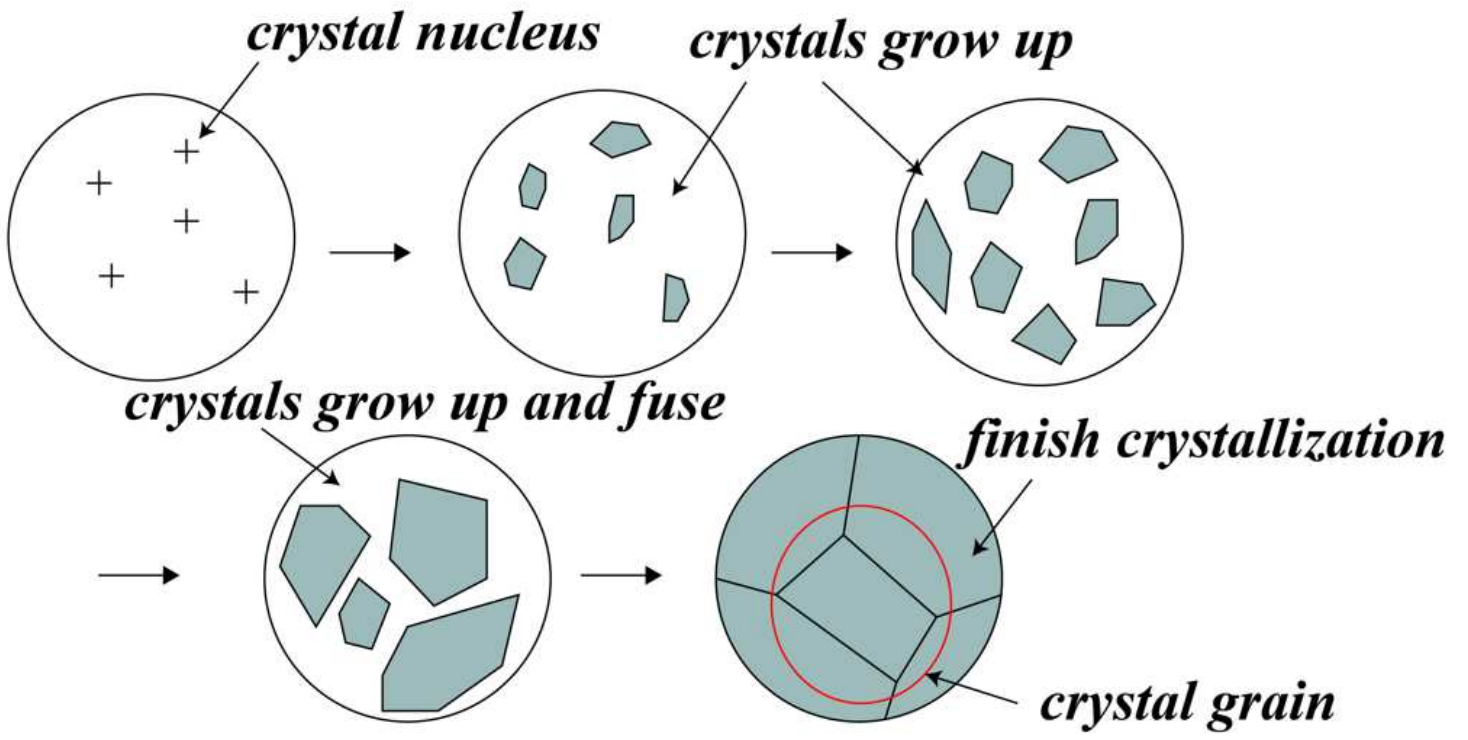


Figure 3

Process of crystallization phenomenon in the nature

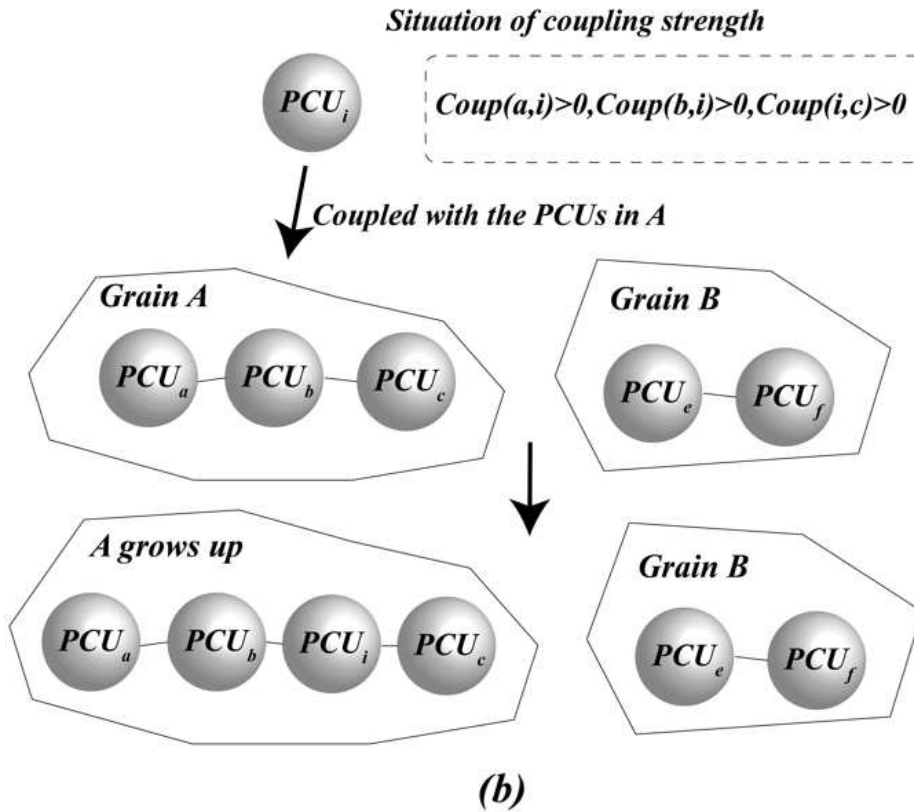
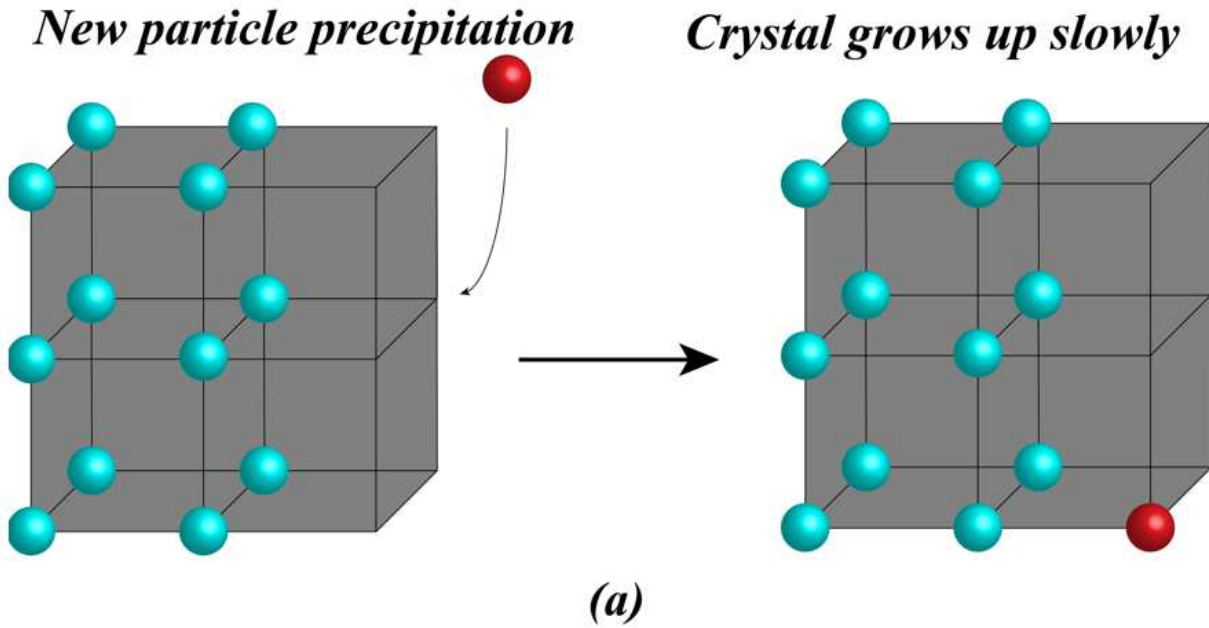


Figure 4

Comparison of crystal growth process. (a) Process of crystal growth in the nature, (b) Process of the PCU-formed crystal growth in configuration design.



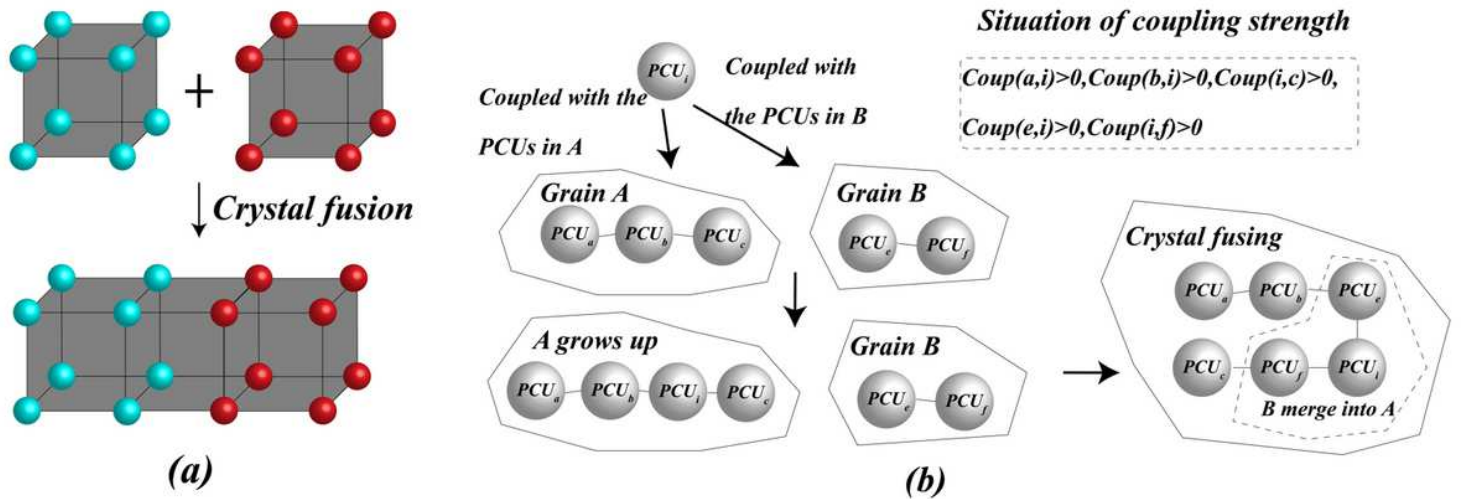


Figure 5

Comparison of crystal fusion process. (a) Process of crystal fusion in the nature, (b) Process of the PCU-formed crystal fusion in configuration design.

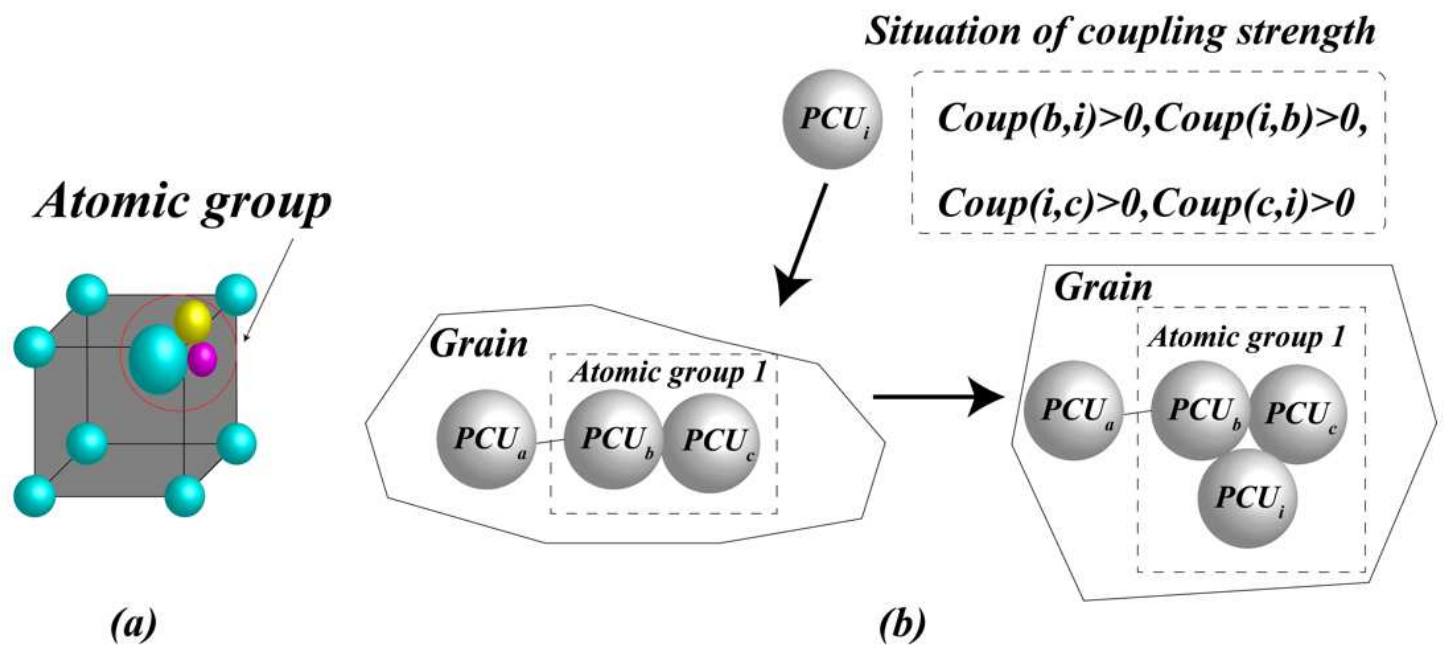


Figure 6

Comparison of atomic group. (a) Atomic group in the nature, (b) Process of the Atomic group formation in configuration design.

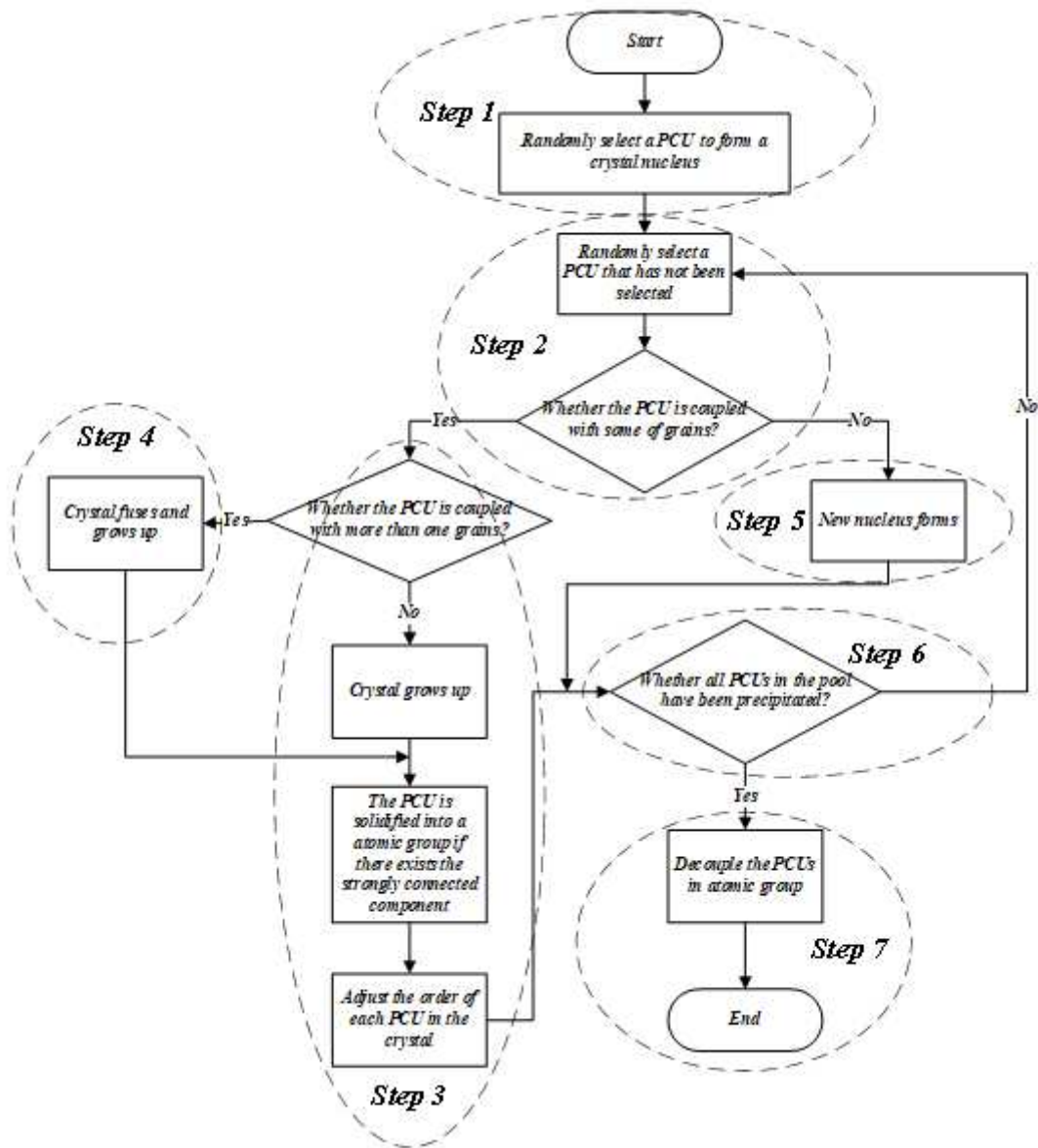
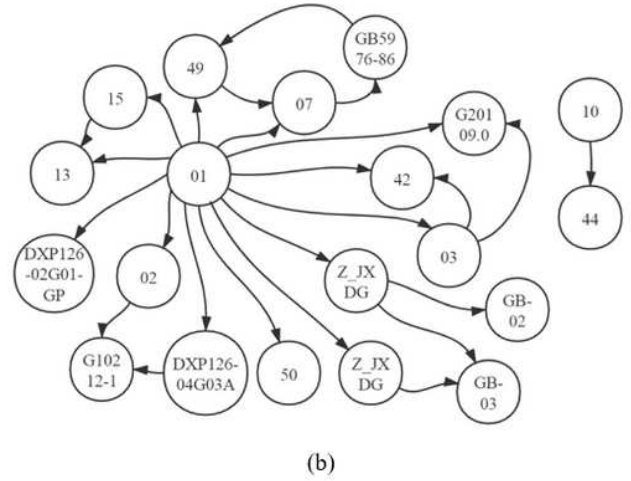
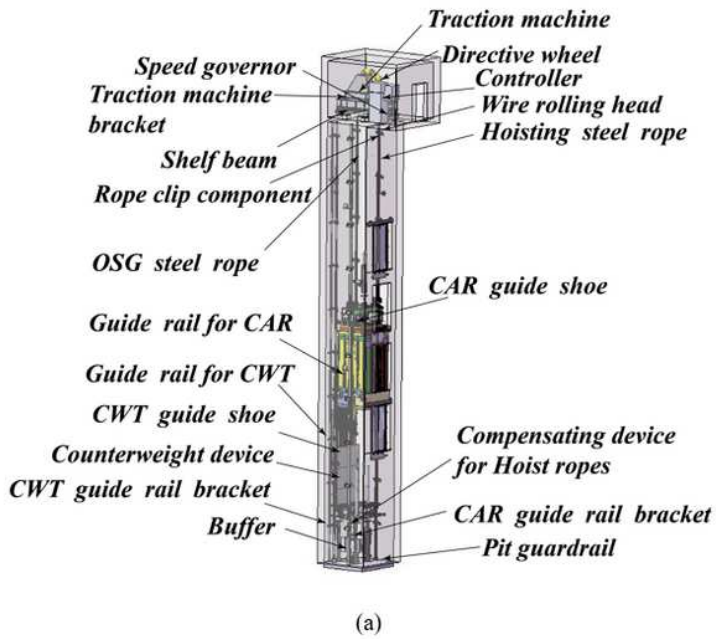


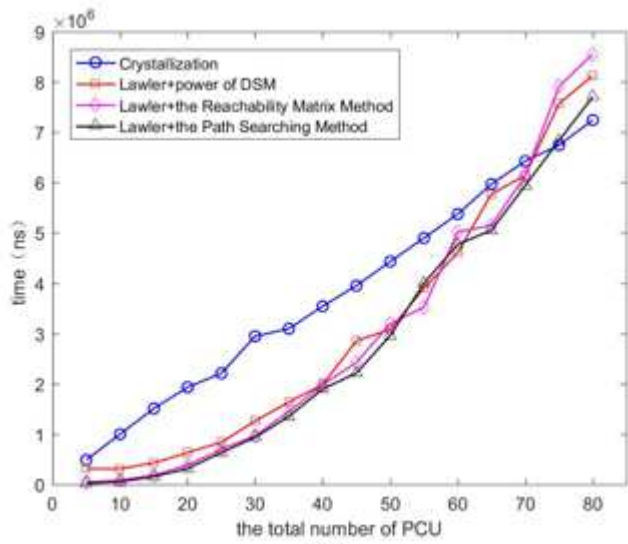
Figure 7

Flowchart of the heuristic method.

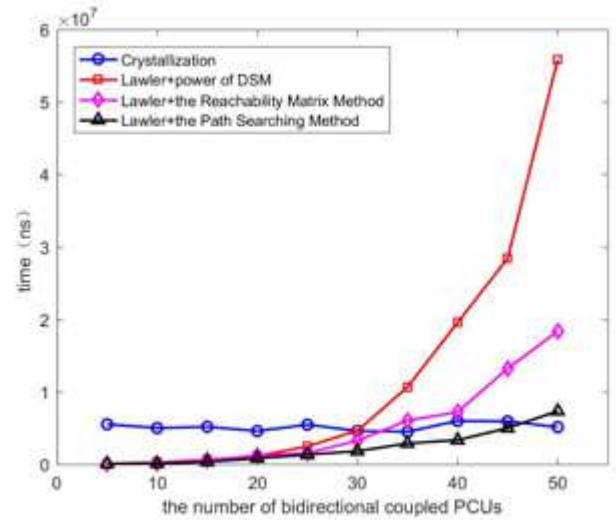


**Figure 8**

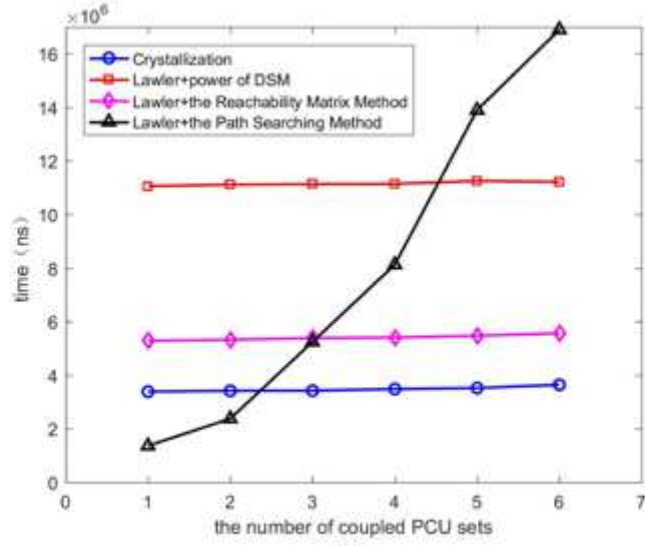
PCU involved in the configuration process. (a) Three-dimensional model of elevator, (b) Directed graph of PCUs.



(a)



(b)



(c)

Figure 9

Running time of each method with different conditions. (a) the total number of PCUs alters, (b) the number of PCUs with bi-directional coupling alters, (c) the number of coupled PCU sets alters.