Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 17 (2014) 622 – 627

www.elsevier.com/locate/procedia

Variety Management in Manufacturing. Proceedings of the 47th CIRP Conference on Manufacturing Systems

Configuration of assembly supply chain using hierarchical cluster analysis

Simon Li[a,]*, Pooya Daie[b]

[a]*University of Calgary, Calgary, Canada*
[b]*Concordia University, Montreal, Canada*

* Corresponding author. Tel.: +1-403-220-5599; fax: +1-403-282-8406. *E-mail address:* simoli@ucalgary.ca

Abstract

The purpose of this paper is to propose a clustering method that configures assembly supply chains. In mass customization, product variety is often characterized by a modular product structure, in which each module may offer several variants to deliver product variants. In this context, assembly supply chain is referred to a network of suppliers who produce and assemble product variants. Given that the proportion of the product variants is specified, the technical problem is how to configure the structure of the assembly supply chain in order to reduce the complexity incurred by product variety. The technical challenge is that the possible number of configurations grows dramatically with the number of product variants. Also, enumerating the possible configurations in a mathematical framework is not a trivial task. Instead of taking an exhaustive approach, the solution approach of this paper is based on hierarchical cluster analysis, in which the tree structure is applied to configure assembly supply chain. The core technique is to investigate the coupling concept in the context of assembly supply chain to characterize the grouping conditions in the structuring process. Few examples are utilized to demonstrate and verify the methodology.

Keywords: assembly supply chain; hierarchical cluster analysis; coupling analysis

1. Introduction

In this paper, product variety is envisioned via product family architecture [1], in which a product is designed in a modular structure, and the variety is achieved by offering several options for each module. A wide range of product variants can then be synthesized by combining different options of each module. Beyond product modularity, Salvador et al. [2] have studied six cases to investigate the relations between product structures and supply chain configurations.

Based on the concept of product family architecture, Wang et al. [3] modeled an assembly supply chain that described how modules were grouped to deliver sub-assemblies for the final assembler. In this context, the configuration of assembly supply chain is influenced by the proportion of product variants to be offered by the final assembler. For instance, in the case of low variety (i.e., few product variants of high proportions), the final assembler may prefer to assemble most of the modules by itself since the difficulty incurred by product variety is not high. Alternatively, in the case of high variety (i.e., many product variants of even proportions), the final assembler may consider to assign the sub-assembly jobs to some suppliers. This approach actually aligns with the postponement concept that the final assembler can focus on the product differentiation at the later production stage [4]. In this context, the research problem of this paper is how to configure the assembly supply chain based on the modular product structure and the proportions of product variants.

The challenges of the configuration problems in manufacturing have been discussed in the review paper by Hu et al. [5]. One fundamental challenge is that the possible number of configurations is numerous even for a small number of elements (e.g., machines or modules). The early relevant works are found in the context of assembly sequencing, including the cut-set method [6] and the algorithm

doi:10.1016/j.procir.2014.01.145

based on the relational model [7]. Webbink and Hu [8] have applied the grouping corollary to generate possible system configurations analytically. By considering the issues of product variety, the complexity measure has been developed to quantify the complexity aspect related to the assembly system supporting product variety [3, 9].

To choose the configurations that can minimize the complexity measures, the basic approach is mainly based on three general steps: (1) generate all possible configurations, (2) compute the complexity measures, and (3) select the configuration of minimum complexity [9]. Some techniques to expedite the solution process include the elimination of the asymmetric configurations [10] and the use of mathematical propositions for guidance [3]. Compared with the configuration efforts in assembly sequences and supply chains, the application of cluster analysis as the solution approach is relatively limited in literature. This paper is intended to contribute in this research direction.

Generally, the complexity issue in the configurations of assembly supply chains stems from the large number of possible configurations. To manage the complexity due to the high number of elements, one intuitive approach is to group the highly-linked elements and identify the structure of the groups. Cluster analysis is basically applied to facilitate the grouping process in the formation of configurations. The specific new concept of this paper is to formulate the coupling values based on the product variety information. Notably, cluster analysis has been recognized as one approach for solving cell formation problems [11].

Nomenclature

$ap_{i,j}$	adjusted coupling value between the ith and jth module options
br_{ij}	binary relation between the ith module option and the jth product variant
C	complexity measure
$cp_{i,j}$	coupling value between the ith and jth module options
e	total number of edges of the assembly supply chain
e_i	number of input edges of the ith supplier
md_i	ith module
$md_{i,j}$	jth option of the ith module
n	total number of product variants
nq_i	normalized mix ratio of the ith module option
p	total number of modules
pv_i	ith product variant
$q_{i,v}$	mix ratio of the ith supplier for producing the vth options or variants
sp_i	ith supplier

2. Background: Assembly Supply Chain and Complexity

To maintain the readability of this paper, we briefly discuss the model of the assembly supply chain and the corresponding complexity measure. For readers who want more details, please refer to [3].

In the problem context, the company plans to produce a product that comes with multiple variants. Let $PV = \{pv_1, pv_2, \ldots, pv_n\}$ be the set of n product variants. To achieve product variety, all product variants share the same modular product structure. Let $MD = \{md_1, md_2, \ldots, md_p\}$ be the set of p modules. Each module can offer more than one option, and let $md_{i,j}$ denote the jth option of the ith module. Figure 1 illustrates three modules, and each module has two options. Correspondingly, Figure 2 illustrates three possible product variants. Notably, the total possible number of product variants in this case is the multiplication of the numbers of module options (i.e., 2*2*2 = 8).

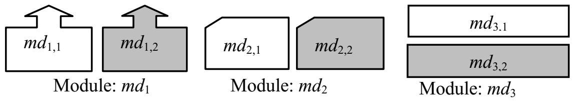

Fig. 1. Illustration of modules and module options.

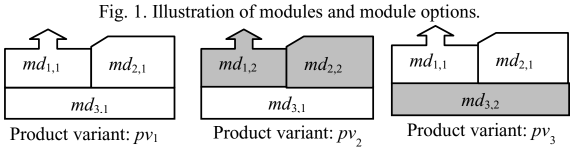

Fig. 2. Illustration of product variants.

With reference to [3], a sample structure of an assembly supply chain is given in Figure 3, in which the node sp_i stands for the ith supplier. There are two special nodes at both ends of the assembly supply chain (i.e., sp_0 and sp_5). The node sp_0 can be viewed as a provider of raw materials, and the node sp_5 is the final assembler who yields the final product variants. Except these two special nodes, other nodes have only one output edge for producing a set of module options or sub-assemblies based on what they receive. To indicate the proportion of different module options or sub-assemblies to be produced from a supplier, a mix ratio will be assigned to each supplier (except sp_0). Let $q_{i,v}$ be the mix ratio of the ith supplier for producing the vth module options, sub-assemblies or product variants. When the mix ratio of the product variants (output of the final assembler) is known, the mix ratios pertaining to the suppliers can be determined. Table 1 lists one example of the mix ratios according to the assembly supply chain demonstrated in Figures 1, 2 and 3.

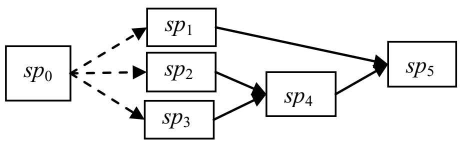

Fig. 3. An example of assembly supply chain.

In assembly supply chain, the complexity measure has been developed based on the concept of information entropy, and the formulation of the complexity measure according to [3] (denoted as C) is given as follow.

$$C = \log_2 e - \frac{1}{e}\sum_i e_i \sum_v q_{i,v} \log_2 q_{i,v} \tag{1}$$

where e_i is the number of input edges of the ith supplier, and e is the total number of edges calculated by $e = \sum e_i$. Given the mix ratio of product variants, the complexity measure is influenced by the structural configuration (i.e., e and e_i) and the mix ratios of suppliers (i.e., $q_{i,v}$). In brief, more edges and even mix ratios lead to higher complexity. Concerning the modularity of the assembly supply chain [9], the non-modular configuration tends to have less number of edges, and the modular configuration provides some opportunity to produce sub-assemblies of "uneven" mix ratios (i.e., high ratio value on particular variants, leading to lower complexity).

Given the model of assembly supply chain and the complexity measure, the technical question is how to configure the assembly supply chain in order to minimize the complexity measure.

Table 1. Example of the information of each supplier.

Supplier	Output	Mix ratio	No. of input edges
sp_1	$\{md_{1,1}, md_{1,2}\}$	$[q_{1,1}; q_{1,2}] = [0.8; 0.2]$	$e_1 = 1$
sp_2	$\{md_{2,1}, md_{2,2}\}$	$[q_{2,1}; q_{2,2}] = [0.8; 0.2]$	$e_2 = 1$
sp_3	$\{md_{3,1}, md_{3,2}\}$	$[q_{3,1}; q_{3,2}] = [0.9; 0.1]$	$e_3 = 1$
sp_4	$\{(md_{1,1}, md_{2,1}), (md_{1,1}, md_{2,2}), (md_{1,2}, md_{2,1}), (md_{1,2}, md_{2,2})\}$	$[q_{4,1}; q_{4,2}; q_{4,3}; q_{4,4}] = [0.8; 0; 0; 0.2]$	$e_4 = 2$
sp_5	$\{pv_1, pv_2, pv_3\}$	$[q_{5,1}; q_{5,2}; q_{5,3}] = [0.7; 0.2; 0.1]$	$e_5 = 2$

3. Methodology

3.1. Concept of coupling and assembly supply chain

The foundation of the methodology is based on hierarchical cluster analysis (HCA). In HCA, one crucial step before the clustering process is to evaluate the similarity (or dissimilarity) between any two objects. The clustering process tends to group the objects that share high similarity values with each other. Since the term "similarity" may imply the common traits between two objects (e.g., both flies and eagles have wings), this paper uses the term "coupling" to imply a more general concept. Particularly, the coupling is referred to the appropriateness of two objects to be grouped in one application. If it is appropriate to group two objects in one application, the corresponding coupling value is high. Notably, the coupling value is application-dependent. That is, we may consider that objects a and b should be grouped in one case but not be grouped in another case. Then, the application context becomes the important information to guide the formulation of coupling values.

In assembly supply chain, the key "coupling" question is whether two modules should be grouped together and to explain why. For example, in Figure 3, should we group md_1 and md_2? If so (or not), what is the reason? As any intermediate supplier (to produce sub-assemblies) will incur more edges (i.e., higher complexity), it is important for an intermediate supplier to yield a "low-variety" mix ratio to counter the additional edges. In view of coupling, if two modules can be combined to yield a "low-variety" mix ratio, the coupling values between these two modules will be high. Based on this idea, the detailed quantification of the coupling values is developed in the methodical procedure.

Notably, the procedure for HCA (e.g., building the dendrogram or tree) has been well documented in literature (e.g., [12]), and it will not be repeated here. The methodical procedure in the next sub-section will focus on how to compute the coupling values for building the tree and how to configure the assembly supply chain based on the tree result.

3.2. Methodical procedure

Step 1: fill in the product variety form

This step is to collect the input information of product variety and fill it in a compact form, namely, the product variety form. This form consists of two major parts. One part is the binary matrix that captures the relations between module options and product variants. The gray area in Figure 4 shows such a matrix according to the example in Figure 2. Let br_{ij} be the binary matrix entry that $br_{ij} = 1$ if the ith module option is selected in the jth product variant (else, $br_{ij} = 0$).

Another part is to record the mix ratios of module options and product variants, which are placed along the corresponding rows and columns, respectively. For illustration, the mix ratios of the example based on Table 1 are also recorded in Figure 4. For normalization, the mix ratios of module options are divided by the number of modules (i.e., 3 in the example), and the normalized value of the ith module option is denoted as nq_i (i.e., the last column of Figure 4).

	pv_1 (mix ratio)	pv_2 (mix ratio)	pv_3 (mix ratio)	Mix ratio of module options	Normalized mix ratio
$md_{1,1}$	$br_{11} = 1$	$br_{12} = 0$	$br_{13} = 1$	$q_{1,1} = 0.80$	$nq_1 = 0.27$
$md_{1,2}$	$br_{21} = 0$	$br_{22} = 1$	$br_{23} = 0$	$q_{1,2} = 0.20$	$nq_2 = 0.07$
$md_{2,1}$	$br_{31} = 1$	$br_{32} = 0$	$br_{33} = 1$	$q_{2,1} = 0.80$	$nq_3 = 0.27$
$md_{2,2}$	$br_{41} = 0$	$br_{42} = 1$	$br_{43} = 0$	$q_{2,2} = 0.20$	$nq_4 = 0.07$
$md_{3,1}$	$br_{51} = 1$	$br_{52} = 1$	$br_{53} = 0$	$q_{3,1} = 0.90$	$nq_5 = 0.30$
$md_{3,2}$	$br_{61} = 0$	$br_{62} = 0$	$br_{63} = 1$	$q_{3,2} = 0.10$	$nq_6 = 0.03$
	$q_{5,1} = 0.70$	$q_{5,2} = 0.20$	$q_{5,3} = 0.10$		

Fig. 4. Product variety form.

Step 2: determine the coupling values of module options

Two module options are coupled if they are selected in the same product variant(s). In the reasoning, if two specific module options are often selected at the same time in many product variants, we can group them together to form a sub-assembly that can be produced with high volume and low variety. Based on the binary matrix in the product variety form, the Jaccard coefficient is applied to evaluate the coupling values. Let $cp_{i,j}$ be the coupling value between the ith and jth module options, and its formulation is given below.

$$cp_{i,j} = \frac{\sum_{k=1}^{n} \min(br_{ik}, br_{jk})}{\sum_{k=1}^{n} \max(br_{ik}, br_{jk})} \tag{2}$$

Recall that n is the total number of product variants. Notably, the min and max operations can be considered as an alternative way to count the 1-1 and 1-0 matches in the Jaccard coefficient. For illustration, Figure 5 shows the square coupling matrix that records the coupling values between two module options of the example.

Step 3: adjust the coupling values based on mix ratios

If the mix ratios of two module options are high, the grouping of these module options can potentially reduce the complexity (for high production of specific variants). Thus, the coupling values computed in the previous step are adjusted based on the information of mix ratios. Let $ap_{i,j}$ be the adjusted coupling value between the ith and jth module options, and its formulation is given below. Figure 6 shows the matrix of adjusted coupling values for the example.

$$ap_{i,j} = cp_{i,j}(nq_i + nq_j) \quad (3)$$

	$md_{1,1}$	$md_{1,2}$	$md_{2,1}$	$md_{2,2}$	$md_{3,1}$	$md_{3,2}$
$md_{1,1}$		0	1	0	0.33	0.50
$md_{1,2}$	0		0	1	0.50	0
$md_{2,1}$	1	0		0	0.33	0.50
$md_{2,2}$	0	1	0		0.50	0
$md_{3,1}$	0.33	0.50	0.33	0.50		0
$md_{3,2}$	0.50	0	0.50	0	0	

Fig. 5. Coupling matrix of module options.

	$md_{1,1}$	$md_{1,2}$	$md_{2,1}$	$md_{2,2}$	$md_{3,1}$	$md_{3,2}$
$md_{1,1}$		0	*0.54*	*0*	0.19	0.15
$md_{1,2}$	0		*0*	*0.14*	0.19	0
$md_{2,1}$	0.54	0		0	0.19	0.15
$md_{2,2}$	0	0.14	0		0.19	0
$md_{3,1}$	0.19	0.19	0.19	0.19		0
$md_{3,2}$	0.15	0	0.15	0	0	

Fig. 6. Adjusted coupling matrix.

Step 4: determine the coupling values between modules

When configuring an assembly supply chain, the grouping process is essentially carried out over the modules rather than module options. Thus, this step is to determine the average of the coupling values between two modules by considering all relevant module variants. For example, by checking Figure 6, there are four coupling values between md_1 and md_2 (italicized for highlight), and we can find the average of these four values to reflect the coupling between md_1 and md_2. Figure 7 shows the corresponding results of the example.

	md_1	md_2	md_3
md_1		0.17	0.13
md_2	0.17		0.13
md_3	0.13	0.13	

Fig. 7. Coupling matrix between modules.

Step 5: construct the tree

Based on the coupling matrix between modules obtained from the previous step, the standard procedure of hierarchical cluster analysis is applied by treating coupling values same as similarity measures. The resulting tree of the example is given in Figure 8.

Step 6: cut the tree to suggest the configuration of the assembly supply chain

The top branch of the tree can be considered the position of the final assembler. Then, the tree structure basically suggests how different modules should be grouped leading to the final assembler. Consider two cut lines in Figure 8 as the example. If the cut line #1 is applied, two branches are cut, indicating that the final assembler receives two sub-assemblies, one from md_1 and md_2 and another from md_3. Similarly, if the cut line #2 is applied, three branches are cut, and the non-modular configuration is obtained. The configurations based on these two cut lines are shown in Figure 9, where sp_1, sp_2 and sp_3 produce md_1, md_2 and md_3, respectively.

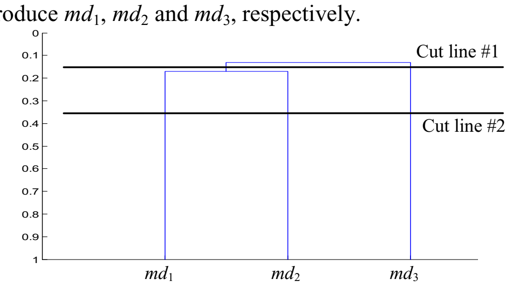

Fig. 8. Tree and cut lines.

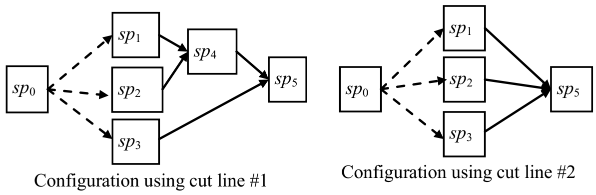

Fig. 9. Configurations and cut lines.

4. Numerical examples

Five numerical cases are set to examine the proposed method of this paper. The characteristics of these cases are discussed as follows.

- Case #1: this case has three modules to yield two product variants. Two module options are used exclusively.
- Case #2: this case has four modules to yield two product variants. Three module options are used exclusively.
- Case #3: this case has three modules, and each module has two variants. It is required to produce all possible product variants with the same mix ratio.
- Case #4: this case has four modules to yield five product variants. No module option is used exclusively.
- Case #5: this case is similar to Case #4 except that some module options are purposely selected with high mix ratios.

To keep the discussion easy to follow, the numerical details and the results of these cases are organized in the following sub-sections.

4.1. Discussion of Cases #1 and #2

The product variety forms of Cases #1 and #2 as the methodical input are provided in Figure 10. While both cases intend to yield two product variants with the same mix ratio (i.e., 0.5), these variants are differentiated only by one module (i.e., md_3 for Case #1 and md_4 for Case #2). Thus, the sensible configurations should group the modules that contribute the common options of all product variants.

The trees and configurations based on the proposed method for Cases #1 and #2 are shown in Figures 11 and 12, respectively. First of all, both trees show that the differentiating modules have lower coupling values with other modules. Following the tree structures, the configurations suggest forming sub-assemblies of the "non-differentiating" modules, and these align with the sensible configurations discussed earlier. Algorithmically, the tree is constructed by grouping two modules at one time. Yet, if three modules have close coupling values with each other, the tree can still yield such a structure approximately, as shown in Case #2.

To compactly represent the configurations, we adapt the "bracket" representation from [3]. For example, the configurations from Figures 11 and 12 can be represented as ((sp_1, sp_2) (sp_3)) and ((sp_1, sp_2, sp_3) (sp_4)). To verify the results, the complexity measure is applied to compare the configurations in Figures 11 and 12 with the non-modular configurations. The complexity measures are provided in Table 2, which shows that the configurations in Figures 11 and 12 have lower complexity.

a)

	pv_1	pv_2	
$md_{1,1}$	1	1	1
$md_{2,1}$	1	1	1
$md_{3,1}$	1	0	0.5
$md_{3,2}$	0	1	0.5
	0.5	0.5	

b)

	pv_1	pv_2	
$md_{1,1}$	1	1	1
$md_{2,1}$	1	1	1
$md_{3,1}$	1	1	1
$md_{4,1}$	1	0	0.5
$md_{4,2}$	0	1	0.5
	0.5	0.5	

Fig. 10. Product variety forms of (a) Case #1 and (b) Case #2.

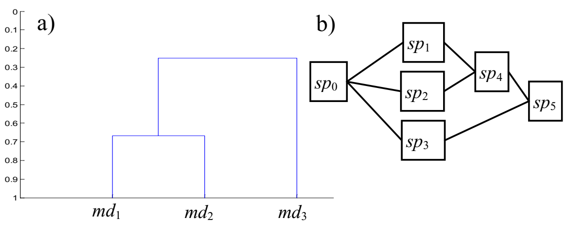

Fig. 11. Results of Case #1 (a) tree and (b) configuration.

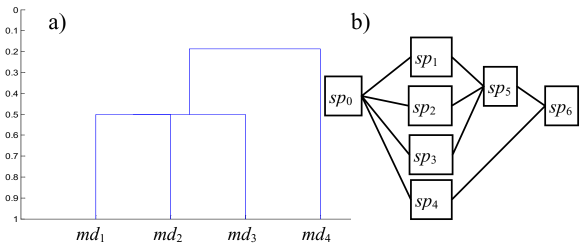

Fig. 12. Results of Case #2(a) tree and (b) configuration.

Table 2. Comparison of Cases #1 and #2 with non-modular configuration.

	Configuration	Complexity
Case #1	((sp_1, sp_2) (sp_3))	3.236
	(sp_1, sp_2, sp_3)	3.252
Case #2	((sp_1, sp_2, sp_3) (sp_4))	3.503
	(sp_1, sp_2, sp_3, sp_4)	3.625

4.2. Discussion of Case #3

Different from Cases #1and #2, Case #3 has equal mix ratios of all possible variants that are constructed from three modules and two module options. Figure 13 shows the product variety form of this case, and Figure 14 shows the resulting tree and configuration. Notably, this is a "non-modular" configuration.

According to Proposition 2 in [3], it is generally suggested that "modular" configurations are preferred for the situations of equal demand shares. In their proof of Proposition 2, the complexity of modular configurations is higher than that of non-modular configurations only if the numbers of modules and variants are high enough (to satisfy a threshold condition). In our brief analysis, a modular configuration tends to yield a higher number of edges, which lead to higher complexity measures. Yet, based on the complexity formulation in (1), the increase of $\log_2 e$ is less steep for large e values. This explains why modular configurations are generally preferred in case of higher numbers of modules and variants. However, the proposed method does not discern the situations of higher numbers of modules and variants (from the lower ones) given the equal demand shares of all possible variants. Further research is required to address this special situation.

	pv_1	pv_2	pv_3	pv_4	pv_5	pv_6	pv_7	pv_8	
$md_{1,1}$	1	1	1	1	0	0	0	0	0.5
$md_{1,2}$	0	0	0	0	1	1	1	1	0.5
$md_{2,1}$	1	1	0	0	1	1	0	0	0.5
$md_{2,2}$	0	0	1	1	0	0	1	1	0.5
$md_{3,1}$	1	0	1	0	1	0	1	0	0.5
$md_{3,2}$	0	1	0	1	0	1	0	1	0.5
	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	

Fig. 13. Product variety form of Case #3.

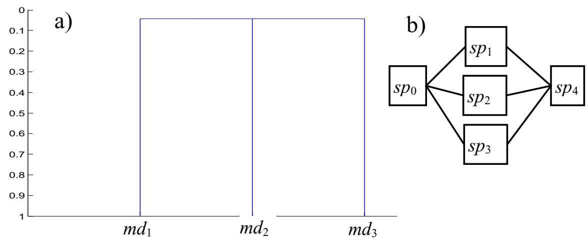

Fig. 14. Results of Case #3 (a) tree and (b) configuration.

4.3. Discussion of Cases #4 and #5

Both Cases #4 and #5 have four modules, and each module has three options to yield five product variants in total. For investigation purpose, we attempt to "randomly" pick up the module options to construct product variants in Case #4. In contrast, some module options are used more frequent than others in Case #5. The product variety forms of Cases #4 and #5 as the methodical input are provided in Figure 15. The trees and configurations based on the proposed method for Cases #4 and #5 are shown in Figures 16 and 17, respectively.

In Case #4, the highest coupling value that joins two modules (i.e., md_2 and md_3) is about 0.07 (see the tree in Figure 16). If we set the cut line of the tree at 0.1, we will obtain the non-modular configuration that the final assembler addresses all modules without sub-assemblies. In contrast, if the cut line is set at 0.1 in Case #5, we obtain a configuration that groups md_1 & md_2 and then with md_3 (see the tree in Figure 17). This configuration generally makes sense because $md_{1,1}$ and $md_{2,1}$ are used together with high mix ratio 0.89.

To further examine the proposed method, we have identified all possible configurations exhaustively and evaluate the complexity measures for Cases #4 and #5. The results are recorded in Table 3, and the complexity measures of the configurations suggested by the proposed method are highlighted in gray color. As seen in Table 3, the proposed method can suggest the configurations of the lowest complexity for Cases #4 and #5. These results support the utility of the proposed method towards the complexity reduction of an assembly supply chain.

a)

	pv_1	pv_2	pv_3	pv_4	pv_5	
$md_{1,1}$	1	1	0	0	0	0.27
$md_{1,2}$	0	0	1	0	0	0.18
$md_{1,3}$	0	0	0	1	1	0.55
$md_{2,1}$	1	0	0	1	1	0.66
$md_{2,3}$	0	1	1	0	0	0.34
$md_{3,1}$	0	0	0	1	0	0.31
$md_{3,2}$	0	1	1	0	0	0.34
$md_{3,3}$	1	0	0	0	1	0.35
$md_{4,1}$	0	1	1	1	0	0.65
$md_{4,2}$	0	0	0	0	1	0.24
$md_{4,3}$	1	0	0	0	0	0.11
	0.11	0.16	0.18	0.31	0.24	

b)

	pv_1	pv_2	pv_3	pv_4	pv_5	
$md_{1,1}$	1	1	1	1	1	1
$md_{2,1}$	1	1	1	1	0	0.89
$md_{2,2}$	0	0	0	0	1	0.11
$md_{3,1}$	1	1	1	0	1	0.69
$md_{3,2}$	0	0	0	1	0	0.31
$md_{4,1}$	1	0	0	1	1	0.66
$md_{4,2}$	0	1	0	0	0	0.16
$md_{4,3}$	0	0	1	0	0	0.18
	0.24	0.16	0.18	0.31	0.11	

Fig. 15. Product variety forms of (a) Case #4 and (b) Case #5.

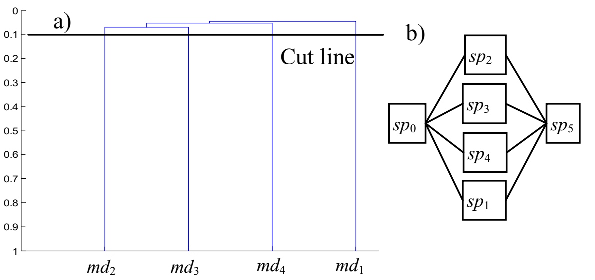

Fig. 16. Results of Case #4 (a) tree and (b) configuration.

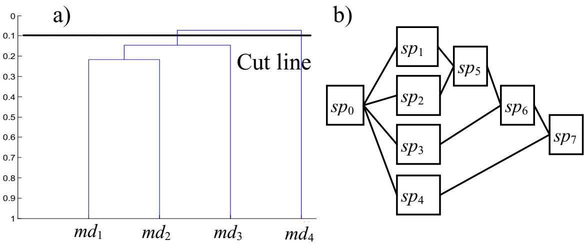

Fig. 17. Results of Case #5 (a) tree and (b) configuration.

5. Closing Remarks

This paper has proposed a method for configuring the assembly supply chain. The method is based on the adaption of hierarchical cluster analysis, and the core technique is to evaluate the coupling according to the product variety information. The proposed method has been examined by five numerical examples, and the preliminary results have demonstrated the utility of the method in view of suggesting sensible configurations and reducing complexity.

Further research is in progress to improve the proposed method. One direction is to address the special situation that all possible product variants are required with equal demand shares. In this case, the concern is to be sensitive to the absolute number of modules in the configuration process. Another direction is to develop a more detailed guideline to cut the tree and suggest the corresponding configuration.

Table 3. Exhaustive comparison of complexity measures in Cases #4 and #5.

Configuration	Case #4	Case #5
(sp_1, sp_2, sp_3, sp_4)	4.767	4.450
((sp_1, sp_2, sp_3) (sp_4))	4.989	4.405
((sp_2, sp_3, sp_4) (sp_1))	4.876	4.708
((sp_1, sp_3, sp_4) (sp_2))	4.989	4.603
((sp_1, sp_2, sp_4) (sp_3))	4.989	4.526
((sp_1, sp_2) (sp_3, sp_4))	5.006	4.519
((sp_1, sp_3) (sp_2, sp_4))	5.115	4.552
((sp_1, sp_4) (sp_2, sp_3))	5.052	4.554
((sp_1, sp_2) (sp_3) (sp_4))	4.868	4.322
((sp_1, sp_3) (sp_2) (sp_4))	4.989	4.409
((sp_1, sp_4) (sp_2) (sp_3))	4.989	4.492
((sp_2, sp_3) (sp_1) (sp_4))	4.843	4.506
((sp_2, sp_4) (sp_1) (sp_3))	4.913	4.587
((sp_3, sp_4) (sp_1) (sp_2))	4.913	4.638
(((sp_1, sp_2) (sp_3)) (sp_4))	5.074	4.401
(((sp_1, sp_2) (sp_4)) (sp_3))	5.074	4.474
(((sp_1, sp_3) (sp_2)) (sp_4))	5.182	4.480
(((sp_1, sp_3) (sp_4)) (sp_2))	5.182	4.598
(((sp_1, sp_4) (sp_2)) (sp_3))	5.182	4.626
(((sp_1, sp_4) (sp_3)) (sp_2))	5.182	4.672
(((sp_2, sp_3) (sp_1)) (sp_4))	5.052	4.567
(((sp_2, sp_3) (sp_4)) (sp_1))	4.984	4.748
(((sp_2, sp_4) (sp_1)) (sp_3))	5.115	4.712
(((sp_2, sp_4) (sp_3)) (sp_1))	5.047	4.821
(((sp_3, sp_4) (sp_1)) (sp_2))	5.115	4.804
(((sp_3, sp_4) (sp_2)) (sp_1))	5.047	4.867

References

[1] Tseng MM, Jiao J, Merchant ME. Design for mass customization. CIRP Annals – Manufacturing Technology 1996; 45: 153-156.

[2] Salvador F, Rungtusanatham M, Forza C. Supply-chain configurations for mass customization. Production Planning and Control 2004; 15: 381-391.

[3] Wang H, Ko J, Zhu X, Hu SJ. Wang H, Ko J, Zhu X, Hu SJ. A complexity model for assembly supply chains and its application to configuration design. ASME Journal of Manufacturing Science and Engineering 2010; 132: 021005.

[4] Feitzinger E, Lee HL, Mass customization at Hewlett-Packard: the power of postponement. Harvard Business Review 1997; 75: 116-121.

[5] Hu SJ, Ko J, Weyand L, ElMaraghy HA, Lien TK, Koren Y, Bley H, Chryssolouris G, Nasr N, Shpitalni M. Assembly system design and operations for product variety. CIRP Annals – Manufacturing Technology 2011; 60: 715-733.

[6] Baldwin DF, Abell TE, Lui M, De Fazio TL, Whitney DE. An integrated computer aid for generating and evaluating assembly sequences for mechanical products. IEEE Transactions on Robotics and Automation 1991; 7: 78-94.

[7] Homem de Mello LS, Sanderson AC. A correct and complete algorithm for the generation of mechanical assembly sequences. IEEE Transactions on Robotics and Automation 1991; 7: 228-240.

[8] Webbink R, Hu SJ. Automatic generation of assembly system solutions. IEEE Transactions on Automation Science and Engineering 2005; 2: 32-39.

[9] Hu SJ, Zhu X, Wang H, Koren Y. Product variety and manufacturing complexity in assembly systems and supply chains, CIRP Annals – Manufacturing Technology 2008; 57: 45-48.

[10] Koren Y, Shpitalni M. Design of reconfigurable manufacturing systems. Journal of Manufacturing Systems 2010; 29: 130-141.

[11] Yin Y, Yasuda K. Similarity coefficient methods applied to the cell formation problem: a taxonomy and review. International Journal of Production Economics 2006; 101: 329-352.

[12] Everitt BS, Landau S, Leese M, Stahl D. Cluster analysis. 5th ed. West Sussex: John Wiley and Sons; 2011.