A Complexity Model for Assembly Supply Chains and Its Application to Configuration Design

Hui Wang *
Department of Industrial and Operations Engineering
University of Michigan, Ann Arbor, MI 48109

Jeonghan Ko
Department of Industrial and Management Systems Engineering
University of Nebraska-Lincoln, Lincoln, NE, 68588

Xiaowei Zhu, S. Jack Hu
Department of Mechanical Engineering
University of Michigan, Ann Arbor, MI 48109

Abstract

A complexity measure for assembly supply chains has been proposed based on Shannon’s information entropy. This paper extends the definition of such a measure by incorporating the detailed information of the supply chain structure, the number of variants offered by each node in the supply chain, and the mix ratios of the variants at each node. The complexity measure is then applied to finding the optimal assembly supply chain configuration given the number of variants offered at the final assembler and the mix ratios of these variants. The optimal assembly supply chain configuration is theoretically studied in two special scenarios: 1) there is only one dominant variant among all the variants offered by the final assembler, and 2) demand shares are equal across all variants at the final assembler. It is shown that in the first scenario where

*Corresponding author; Address: Department of Industrial and Operations Engineering, University of Michigan, 1205 Beal Avenue, Ann Arbor, MI 48109, USA; phone: 734-764-9103; e-mail: johnwang@umich.edu
one variant dominates the demand, the optimal assembly supply chain should be non-modular; but in the scenario of equal demand shares, a modular supply chain is better than non-modular one when the product variety is high. Finally a methodology is developed to find the optimal supply chain with/without assembly sequence constraints for general demands.

1 Introductions

The product variety offered by many industries has increased greatly in the past few decades. For example, the number of distinct vehicle models in the US increased from 44 in 1969 to 165 in 2005 [1,2]. Another example is the number of styles of running shoes, which increased from 5 in early 70s to 285 in the late 90s [3]. Such increases were motivated by the desire to provide high product variety and highly customized products in response to fierce competition in the market. However, high variety brings a lot of challenges to manufacturing. Several studies have shown that high product variety has negative impact on manufacturing system performance, such as increasing manufacturing complexity, degrading quality, lowering productivity [4,5].

In order to cope with the challenges caused by high variety, manufacturers implement modular designs in their products. A modular design decomposes the product into several different modules with standard interface. High product variety is achieved through the combinational assembly of different modules. Each module still keeps the high production volume, through which the economy of scale can be maintained.

An important trend enabled by modular products is that manufacturers move from non-
Manufacturers move from non-modular assembly supply chains to modular assembly supply chains, as shown in Figure 1. In a modular assembly supply chain, the manufacturer apportions the product into different sub-assemblies, each of which is obtained through the assembly of several modules. Some of the sub-assemblies are outsourced to the suppliers and the manufacturer only does the final assembly of a few sub-assemblies. For example, Volvo’s S80 model is assembled from 23 different sub-assemblies, delivered to the final assembly line through 17 pre-assembled units, 11 of which are operated by the suppliers [6].

When a manufacturer wants to introduce a new modular product into the market, there are two important decisions that need to be made in the design stage: 1) how to apportion the product into different modules; 2) how many variants to offer in each module. These two decisions will then determine product variety, i.e., the number of variants in the final assembled product, offered by the manufacturer to customers. Based on that product variety, the manufacturing system layout and assembly supply chain configuration can be selected by analyzing and comparing the cost of different systems and supply chain configurations. However, cost analysis of assembly supply chains requires the estimation of many parameters.
such as manufacturing cost, overage and shortage cost, transportation and production lead time, etc. Some of these parameters are difficult to estimate in practice. In addition, sophisticated cost models bring a lot of analytical challenges due to the network structure of an assembly supply chain, especially when the product variety is taken into consideration. That is why most research is based on empirical case studies when studying the effect of product variety on supply chain configurations [7,8].

A new complexity measure for assembly supply chains has been proposed recently by Wang et al. [9]. This complexity measure is based on Shannon’s information entropy and takes the following factors into consideration: the supply chain structure, product variety level of each node and the mix ratios of variants offered by each node. In addition, Wang et al. [9] investigated the relationship between the complexity and cost of an assembly supply chain and showed that the cost and complexity agree with each other when comparing assembly supply chains with the same level of product variety, but different configurations. It is shown that the decisions made by the complexity and cost criterion follow the same trend and more specifically, both complexity and cost, based on configuration selection, shift their preference from non-modular assembly supply chains to modular assembly supply chains, as product variety increases.

This paper extends the complexity model to incorporate more detailed information on supply chain structure and product variety. Then the complexity model is applied to the selection of optimal assembly supply chain configuration, given the number of variants offered by the manufacturer (the final assembler in the assembly supply chain). The paper is organized as follows. In Section 2, we review the relevant literature. In Section 3, we derive the complexity model based on Shannon’s information entropy by incorporating detailed...
information on supply chain structure and product variety. In Section 4, we theoretically investigate the supply chain configuration selection problem in two special scenarios. In Section 5, we study the supply chain selection problem in general demand scenario. We first develop a decomposition iterative algorithm to generate all possible supply chain candidates without assembly sequence constraints. Then, we extend the results and develop the methodologies to find the optimal assembly supply chain when product assembly sequence constraints exist. Section 6 gives some further discussion about this complexity research and Section 7 concludes the paper.

2 Literature Review

Assembly supply chain is an important research area in supply chain management and most research in this area is based on inventory management and cost analysis. Early research of assembly supply chains focuses on centralized assembly systems, where a single decision-maker determines the ordering policies of all members in the supply chain to minimize the total cost. Schmidt and Nahmias [10] investigated a finite-horizon model of a centralized assembly supply chain with two components. Rosling [11] investigated the periodic review, infinite-horizon inventory problem in assembly supply chains under random demands. Recently decentralized assembly supply chains have gained more and more attention from researchers. In a decentralized assembly supply chain, each member in the supply chain makes the decision based on the decisions made by its upstream suppliers and downstream assembler to minimize its own cost. Wang and Gerchak [12] studied a decentralized assembly supply chain with perfect component yield and stochastic demand for the end product,
where the suppliers make capacity decisions for the production of components. Bernstein and DeCroix [13] studied equilibrium price and capacity decisions in a multiple-echelon modular assembly supply chain and showed that the modular assembly supply chain could be more beneficial than the non-modular assembly supply chain when the introduced sub-assemblers have lower assembly cost than the final assembler. However, most research in this area focuses on assembly supply chains of single product. In this paper, we incorporate product variety into the assembly supply chain model and investigate assembly supply chains using a new performance measure, complexity.

A few papers address how product variety influences supply chain configuration decision. Randall and Ulrich [7] used data from U.S. bicycle industry to examine the relationship among product variety, supply chain structure and system performance. It was shown that firms that match their supply chain structure to the type of variety they offer often outperform those that fail to make such choices. Salvador et al. [8] used empirical data to explore how a firm’s supply chain, defined as the whole of its supply, manufacturing and distribution networks, should be configured, when different degrees of customization are offered. We focus on a similar problem in the context of assembly supply chains, but instead of applying empirical case study, we use a model-based method to find the solution.

Several different definitions of complexity have been proposed by researchers within different disciplines. For example, Suh [14] defined a complexity measure particularly applicable in product design; Cover and Thomas [15] discussed Kolmogorov complexity, which is a measure of computational resources needed to describe a string of text. A commonly-used complexity definition is based on the information entropy, proposed by Shannon [16] in the context of communications systems. Shannon’s information entropy is a measure of
the uncertainty surrounding the outcome of a random experiment. Suppose we have a random experiment with \( n \) possible outcomes, whose probabilities of occurrence are \( q_1, q_2, \ldots, q_n \). Then the information entropy of this random experiment, showing how uncertain we are of the outcome, is of the following form,

\[
H = -K \sum_{i=1}^{n} q_i \log_2 q_i
\]

(1)

where the positive constant \( K \) merely amounts to a scaling factor.

Information entropy plays an important role in communication systems and other disciplines as a measure of information, choice and uncertainty. It has been used to study complexity in many different areas, including biology, physics, manufacturing systems, supply chains, etc [17,18]. For example, Deshmukh et al. [19] derived an information-theoretic entropy measure of complexity for a given combination and ratio of part types to be produced in a manufacturing system. Zhu et al. [20] studied the operator choice complexity in mixed-model assembly lines and developed a methodology to find the optimal assembly sequence to minimize manufacturing complexity. Sivadasan et al. [21] developed an experimental methodology to study the operational complexity in a supplier-customer system. Following the similar method, Frizelle [22] developed a metric to measure the complexity within the supply chain, called operational complexity. Wang et al. [9] proposed a complexity measure for assembly supply chains in the presence of product variety and studied the relationship between complexity and cost of an assembly supply chain. Based on the complexity model, Wang et al. [23] studied how to make the decision of assembly supply chain configuration under different product variety level. In this paper we extend the complexity model of Wang et al. [9] by incorporating more detailed information on supply chain structure and product
variety. Based on this complexity measure, we study the assembly supply chain selection problem given the product variety.

3 Complexity Model of Assembly Supply Chains

In this section, we derive the complexity model based on Shannon’s information entropy by incorporating the detailed information on the supply chain structure, the number of variants each node produces and the mix ratios of the variants offered by each node. Here we follow the traditional definition of an assembly supply chain (also called assembly system in supply chain literature) in the context of supply chain management, where each node can have multiple upstream suppliers, but a given node can only supply one downstream node. Suppose a general assembly supply chain takes the form in Figure 2. The final product is apportioned into \( m \) different modules and each module has several different variants. There are \( n \) nodes in the supply chain and each node can produce one product (either one module, such as node 1 and 2, or one sub-assembly from the assembly of several modules, such as node \( m + 1 \)) with certain number of variants. We assume that a downstream node can assemble any combination of the variants provided by its upstream suppliers, and each combination counts as a distinct variant. This assumption can easily be relaxed by setting the demand of the non-existing variants as zero. Suppose only one upstream node provides all the variants of one particular component for its downstream assembler. Following this assumption, we know that the number of nodes in the most upstream echelon is equal to the number of modules in the final product, \( m \). Here we number the nodes in the following sequence by convention. Node 1, 2, \ldots, m are the nodes in the most upstream echelon. Node
Figure 2: A general assembly supply chain and relationships of variants’ demand share

$m + 1, \ldots, n - 1$ are intermediate sub-assemblers and node $n$ is the final assembler. Since the nodes in the most upstream echelon do not have suppliers and we want to capture all the supply-assembly activities in the whole supply chain, a virtual supplier is introduced here, denoted as node 0, which supplies all the raw materials to nodes in the most upstream echelon. Notice that this virtual supplier, node 0, is a special node in the supply chain because it has multiple downstream nodes.

As regards the number of variants and their mix ratios, we define the following notations:

$V_i$: the number of variants that node $i$ can produce, $i = 1, \ldots, n$.

$S_i$: the set of nodes that are suppliers to node $i$, $i = 1, \ldots, n$.

$A_{iju}$: the set of variants produced at node $i$ that use variant $u$ from node $j$, where node $j$ is a supplier to node $i$, i.e., $j \in S_i$.

At each time period, one unit product is demanded at node $i = 1, \ldots, n - 1$ from its downstream assembler and for the final assembler, one unit product is demanded by the
customer. Assume the demands of the variants at node $i$ are independent, $i = 1, \ldots, n$. Let $q_{iv}$ denote the probability that at each time period, the variant $v = 1, \ldots, V_i$ is demanded at node $i$, which is also equal to the fraction of node-$i$ demand that belongs to the variant $v$ in the long run. Hereafter, we refer to $q_{iv}$ as the demand share of variant $v$ at node $i$.

In addition, define the vector $q_i := (q_{i1}, q_{i2}, \ldots, q_{iV_i})$, which captures the mix ratios of the variants produced by node $i$. Hereafter, we refer to $q_i$ as the demand vector of node $i$.

The final assembler’s demand vector, $q_n$, determines how the demand at upstream nodes is allocated among the variants produced by those nodes (see node 1 and 2 and $m + 1$ in Figure 2 for an illustration). Using the notation introduced so far, we have the following relationship between the demand share of variant $u$ at node $j$, $q_{ju}$, and the demand vector of node $i$, $q_i$, where node $j$ is a supplier to node $i$, (i.e., $j \in S_i$):

$$q_{ju} = \sum_{v \in A_{iju}} q_{iv}, j \in S_i.$$  

(2)

The complexity of an assembly supply chain is caused by the following factors: the supply chain structure, product variety level of each node in the supply chain, and demand uncertainty faced by each node. The supply chain structure is determined by the number of nodes in the supply chain and their supply-assembly relationships. The product variety level of a node is the number of variants produced by this node. The demand uncertainty a node faces is decided by the mix ratios of the variants at that node, which tells us the probability that the next demanded unit will be a certain variant. It follows the analogy of information entropy concept: the more evenly distributed the variants of a node, the higher uncertainty of the demand the node faces. Therefore the complexity of an assembly supply chain is determined by and should increase with the following factors: 1) the number of
nodes in the supply chain and their supply-assembly relationships; 2) the number of variants produced by each node in the supply chain; 3) the evenness level of demand mix ratio across all the variants offered by a node in the supply chain.

Following the above argument and the information entropy concept, the complexity model of an assembly supply chain shown in Figure 2, is developed through the following five steps:

**Step 1:** The number of nodes in the supply chain and their relationships are represented by an adjacency matrix, \( \Phi \):

\[
\Phi = \begin{pmatrix}
\phi_{00} & \phi_{01} & \cdots & \phi_{0n} \\
\phi_{10} & \phi_{11} & \cdots & \phi_{1n} \\
\vdots & \vdots & \ddots & \vdots \\
\phi_{n0} & \phi_{n1} & \cdots & \phi_{nn}
\end{pmatrix}
\]

The number of columns and rows of matrix \( \Phi \) equals to the total number of nodes in the supply chain including the virtual supplier, \( n + 1 \). With regard to the supply-assembly relationships, if node \( j \) is a supplier of node \( i \) (i.e., \( j \in S_i \)), then \( \phi_{ji} = 1 \); otherwise, \( \phi_{ji} = 0 \), where \( i = 0, \ldots, n \) and \( j = 0, \ldots, n \).

**Step 2:** For every supply-assembly relationship, where \( \phi_{ji} = 1 \) in matrix \( \Phi \), matrix

\[
Q^{ji} = ( ( q_{ji}^{uv} ) ) = \begin{pmatrix}
q_{00}^{ji} & \cdots & q_{1V_i}^{ji} \\
\vdots & \ddots & \vdots \\
q_{V_j0}^{ji} & \cdots & q_{V_jV_i}^{ji}
\end{pmatrix}
\]

is used to represent the number of variants of node \( j \) and node \( i \) and the mix ratio of the variants of downstream node \( i \). The number of columns of matrix \( Q^{ji} \) is the number of variants produced at node \( i \), \( V_i \) and the number of rows is the number of variants offered by
node \( j \), \( V_j \). Here, \( q_{uv}^{ji} \), \( u = 1, \ldots, V_j \), \( v = 1, \ldots, V_i \) is the production volume of variant \( u \) that node \( j \) needs to produce for the production of variant \( v \) at node \( i \) in order to satisfy the final demand from the customer, assuming that the total demand for all the variants offered by the final assembler is 1.

**Step 3:** Every matrix \( Q^{ji} = (q_{uv}^{ji}) \) is normalized by

\[
\tilde{q}_{uv}^{ji} = \frac{q_{uv}^{ji}}{K} \quad \text{where} \quad K = \sum_i \sum_j \sum_u \sum_v q_{uv}^{ji}
\]

**Step 4:** The assembly supply chain is regarded as a system with the states, whose occurrence probability is \( \tilde{q}_{uv}^{ji} \). Based on Shannon’s information entropy formulation in (1), we define the complexity contribution of any supply-assembly relationship in the supply chain (\( \phi_{ji} = 1 \) in matrix \( \Phi \)), as follows:

\[
C_{ji} = -\sum_u \sum_v \tilde{q}_{uv}^{ji} \log_2 \tilde{q}_{uv}^{ji}
\]

**Step 5:** The complexity of an assembly supply chain is obtained by summing the complexity contribution of all supply-assembly relationships in the supply chain and takes the following form:

\[
C = \sum_i \sum_j C_{ji}
\]

The complexity of an assembly supply chain defined by Equation (5) can be calculated by a simpler formulation (6), where \( L_i \) is the number of suppliers of node \( i = 1, 2, \ldots, n \), and \( K = \sum_{i=1}^n L_i \) is the total number of arcs in the supply chain network,

\[
C = -\sum_{i=1}^n L_i \sum_{v=1}^{V_i} \frac{q_{iv}}{K} \log_2 \left( \frac{q_{iv}}{K} \right)
\]

Wang et al. [9] also interprets this complexity measure through the following random experiment. Suppose for each node, we form a pool of variants produced by that node,
Figure 3: The example used to illustrate the calculation of assembly supply chain complexity where each variant is represented in a quantity proportional to its demand share. Consider the random experiment where we first pick an arc of the supply chain at random, and we then pick one item from the pool of this arc’s end-node. The probability of picking a certain arc, which connects node $i$ to a supplier node $j \in S_i$, and then picking variant $v$ from the pool of node $i$ is $q_{iv}/K$ (since we pick an arc with probability $1/K$ and variant $v$ of node $i$ with probability $q_{iv}$). Wang et al. [9] showed that the information entropy of this random experiment is given by Equation (6). Loosely speaking, the complexity measure indicates the level of uncertainty about the next flow of material that will occur in the supply chain.

An example is given here to illustrate how to calculate the complexity of an assembly supply chain. Suppose one assembly supply chain takes the form as Figure 3. There are five nodes in the supply chain, where node 1, 2, 3 are in the most upstream echelon, node 5 is the final assembler and node 4 is the intermediate sub-assembler. There is one virtual supplier, node 0, which provides all the raw materials to the nodes in the most upstream echelon. Each node provides a certain number of variants and assembles all the possible combinational variants provided its suppliers. Suppose the demand vector of the final assembler is $q_5 =$
\( \left( \frac{1}{2}, \frac{1}{8}, 0, \frac{1}{4}, 0, 0, \frac{1}{8}, 0 \right) \). Based on the demand vector of the final assembler, the demand vector of other nodes in the supply chain can be obtained by Equation (2) and then complexity of this assembly supply chain can be obtained by the following five steps.

Firstly, matrix \( \Phi \) is obtained as follows:

\[
\Phi = \begin{pmatrix}
\phi_{00} & \phi_{01} & \phi_{02} & \phi_{03} & \phi_{04} & \phi_{05} \\
\phi_{10} & \phi_{11} & \phi_{12} & \phi_{13} & \phi_{14} & \phi_{05} \\
\phi_{20} & \phi_{21} & \phi_{22} & \phi_{23} & \phi_{24} & \phi_{25} \\
\phi_{30} & \phi_{31} & \phi_{32} & \phi_{33} & \phi_{34} & \phi_{35} \\
\phi_{40} & \phi_{41} & \phi_{42} & \phi_{43} & \phi_{44} & \phi_{45} \\
\phi_{50} & \phi_{51} & \phi_{52} & \phi_{53} & \phi_{54} & \phi_{55}
\end{pmatrix}
\]

\[= \begin{pmatrix}
0 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

Secondly, for any supply-assembly relationship (i.e., \( \phi_{ji} = 1 \) in matrix \( \Phi \)), matrix \( Q^j \) is developed. For example, we have the following matrix \( Q^45 \) for \( \phi_{45} = 1 \),

\[
Q^{45} = \begin{pmatrix}
\frac{1}{2} & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{8} \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

Similarly, we get \( Q^{01}, Q^{02}, Q^{03}, Q^{14}, Q^{24}, Q^{35} \). Thirdly, matrix \( Q^j \) is normalized by Equation (3), where \( K = 7 \) in this example. For instance, we normalize \( Q^{45} \) and obtain the following normalized matrix \( \tilde{Q}^{45} \),

\[
\tilde{Q}^{45} = \begin{pmatrix}
\frac{1}{14} & \frac{1}{56} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{28} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{56}
\end{pmatrix}
\]
Following the same procedure, we get other normalized matrixes, $\tilde{Q}^{01}$, $\tilde{Q}^{02}$, $\tilde{Q}^{03}$, $\tilde{Q}^{14}$, $\tilde{Q}^{24}$, $\tilde{Q}^{35}$.

Fourthly, the complexity contribution of each supply-assembly relationship, based on each normalized matrix $\tilde{Q}^{ji}$, is calculated through Equation (4). In this example, we calculate the complexity contribution of each relationship as follows: $C_{01} = 0.479, C_{02} = 0.537, C_{03} = 0.537, C_{14} = 0.587, C_{24} = 0.587, C_{35} = 0.651, C_{45} = 0.651$. Finally the complexity of the assembly supply chain is calculated by summing the complexity contribution of all supply-assembly relationships in the supply chain, i.e, $C = C_{01} + C_{02} + C_{03} + C_{14} + C_{24} + C_{35} + C_{45} = 4.029$.

4 Configuration Selection for Assembly Supply Chains in Two Special Scenarios

In this paper, we apply the complexity measure to solving the following decision-making problem of assembly supply chain configuration: After the manufacturer (the final assembler) decides the number of variants offered to the customer and supposedly the demand share of these variants can be estimated, then how should the assembly supply chain be configured so that the supply chain complexity is minimized? Specifically, we want to compare all the possible supply chain configurations in terms of complexity and select the one with the least complexity value as the optimal supply chain configuration. In this section, we theoretically investigate this configuration selection problem in the following two special scenarios: 1) there is one dominant variant among all the variants offered by the final assembler, and 2) demand shares are equal across all variants at the final assembler. In the scenario of
one dominant variant, we show that the optimal supply chain configuration should be non-modular assembly supply chain if the dominance of that variant is big enough. In the scenario of equal demand shares, we show that modular assembly supply chains are more beneficial than non-modular ones when the product variety is high.

Let us consider the first scenario. Suppose we are given the number of variants offered by the final assembler and the mix ratios of these variants. In addition among all the variants offered by the final assembler to customers, there is one dominant variant, which is preferred by most customers and whose demand share is much larger than the demand share of other variants. The following proposition tells us an important property of complexity measure for assembly supply chains when the demand share of that dominant variant is big enough and it provides an useful guideline in how to configure the assembly supply chain in the scenario of one dominant variant.

**Proposition 1.** Suppose among all $V_n$ variants offered by the final assembler, there exists one particularly dominant product, denoted as variant 1 for notation convenience, whose demand share, $q_{n1}$, is much bigger than other variants. If the demand share of that dominant variant increases and approaches 1, i.e, $q_{n1} \to 1$, then

a.) the complexity of the assembly supply chain in Figure 2 only depends on the supply chain structure and equals to $\log_2 K$, where $K$ is the total number of arcs in the supply chain, including the arcs from the virtual supplier to the nodes in the most upstream echelon;  

b.) the optimal assembly supply chain configuration should be non-modular assembly supply chain.

Proposition 1 tells us that given one dominant variant at the final assembler, if the
demand share of that dominant variant is big, we can compare the complexity of different supply chain configurations by simply comparing the number of arcs in different supply chains and then the optimal configuration is just the one with least number of arcs in the supply chain. It also tells us that in the scenario of one dominant variant, introducing intermediate sub-assemblers increases the complexity of the assembly supply chain, which implies that the optimal supply chain configuration should be non-modular.

Now let us consider the second scenario. Suppose the number of variants offered by the final assembler and the mix ratios of these variants are given. Furthermore, we know that customers have equal preference to all variants offered by the final assembler, which means the demand shares are the same across all the variants of the final assembler. Therefore, by Equation (2), the demand shares are also identical across the variants provided by all other nodes in the supply chain. The following proposition can provide some useful guidance in the decision of the supply chain configuration in the scenario of equal demand shares.

**Proposition 2.** Suppose demand shares are equal across all the variants provided by the final assembler, i.e., \( q_n = \left( \frac{1}{V_{n}}, \ldots, \frac{1}{V_{n}} \right)_{1 \times V_n} \). For ease of exposition, we assume all the nodes in the most upstream echelon, node \( 1, \ldots, m \), provide the same number of variants, i.e, \( V_1 = \ldots = V_m = V \) and \( V_n = V^m \). Then if the number of variants produced at each node in the most upstream echelon, \( V \), is high, modular assembly supply chains are more beneficial than non-modular.

Proposition 2 tells us that in the scenario of equal demand shares, modular assembly supply chains could be more preferable than non-modular ones if product variety level is high. But it does not give us specific details about how to configure the modular assembly
supply chain, such as how many number of echelons and intermediate sub-assemblers we should have, how we should allocate the upstream suppliers to the sub-assemblers, etc. The following proposition provides some useful tips in comparing different modular assembly supply chains under the scenario of equal demand shares.

**Proposition 3.** Suppose the demand shares are equal across all the variants provided by the final assembler, i.e., \( q_n = (\frac{1}{V_n}, \ldots, \frac{1}{V_n})_{1 \times V_n} \). In addition, we assume all the nodes in the most upstream echelon, node 1, \ldots, m, provide the same number of variants, i.e, \( V_1 = \ldots = V_m = V \) and \( V_n = V^m \).

a.) Consider the following two modular assembly supply chains. Both of them have one middle intermediate echelon. Supply chain I has \( c \) intermediate sub-assemblers, each of which have \( d \) suppliers from the most upstream echelon; supply chain II has \( d \) intermediate sub-assemblers, each of which have \( c \) suppliers from the most upstream echelon. When the number of variants at the node in the most upstream echelon, \( V \), is big, then the supply chain with larger number of intermediate sub-assemblers has less complexity value than another one, which means if \( c \geq d \), then complexity of supply chain I is less than supply chain II.

b.) Consider the following two modular assembly supply chains. Both of them have one middle intermediate echelon and \( c \) intermediate sub-assemblers in the middle echelon. In supply chain I, every intermediate sub-assembler has the identical number of suppliers from the most stream echelon, \( d \), i.e, \( L_{m+1} = \ldots = L_{m+c} = d \). But in supply chain II, the intermediate sub-assemblers do not have the same number of suppliers from the most stream echelon, i.e, \( L_{m+i} = d + e_i, e_i \in Z, \sum_{i=1}^{c} e_i = 0, i = 1, \ldots, c \). Then the complexity of supply chain I is lower than supply chain II.
Given the equal demand shares of variants at the final assembler, suppose we decide to implement modular assembly supply chains due to high product variety and choose to have one intermediate echelon for sub-assemblers. Then proposition 3 provides some detailed insights regarding how to configure the modular assembly supply chain. If all sub-assemblers in one modular assembly supply chain have the same number of suppliers from the most upstream echelon, we call it a balanced modular assembly supply chain. Similarly, an unbalanced modular assembly supply chain is a modular supply chain, whose sub-assemblers in the middle echelon have different number of suppliers. Proposition 3 tells us under the scenario of equal demand shares, if the number of sub-assemblers in the intermediate echelon is fixed, complexity of the balanced supply chain is lower than the corresponding unbalanced ones. But for two balanced modular supply chains with different number of sub-assemblers in the intermediate echelon, the supply chain with larger number of sub-assemblers is less complex. Following proposition 3, we can easily compare the complexity of the three configurations of Figure 4 and obtain the following complexity relationship, if the demand shares are the same for all the variants offered by the final assembler, \( \text{Complexity}(I) > \text{Complexity}(II) > \text{Complexity}(III) \).
5 Optimal Assembly Supply Chain Selection

In section 4, we discussed how to make the configuration decision of assembly supply chains in two special scenarios: one dominant variant scenario and equal demand shares scenario. But very often the customer demand at the final assembler can’t be classified into either of these two special scenarios. Therefore in this section we study how to find the optimal assembly supply chain under general demands, if we only know the number of variants offered at the final assembler and the mix ratios of these variants. We first develop a decomposition iterative algorithm to generate all possible supply chain candidates without assembly sequence constraints and the optimal one can be obtained through comparing the complexity values of these candidates. Then, we extend the results and develop the methodologies to find the optimal assembly supply chain when product assembly sequence constraints exist.

5.1 Optimal Assembly Supply Chain Selection without Assembly Constraints

For easy explanation, a specific example shown in Figure 5, is used to illustrate the procedure. In this example, a modular product is apportioned into 4 modules, module A, B, C and D, with three, two, one and two variants respectively. Under the assumption of full combinational assembly, the final assembler produces \(3 \times 2 \times 1 \times 2 = 12\) different variants. Suppose the demand share of these 12 variants is estimated as \(d_j, j = 1, \ldots, 12\). Based on these conditions, we want to find the optimal assembly supply chain that has the minimum complexity. Since one upstream node provides all the variants of one particular module (or
Figure 5: The example used to illustrate the methodology of finding the optimal assembly supply chain

sub-assembly) for its downstream assembler, then all the assembly supply chain candidates should have the same number of nodes in the most upstream echelon and the same demand vector at these nodes. In addition, the number of nodes in the most upstream echelon is equal to the number of modules in the product. In this specific example, there will be four nodes in the most upstream echelon shown in Figure 5. Because each node in the supply chain provides the unique product with several variants, in this section we use the name of the product produced at a node as the index of that node for ease of exposition, instead of using number as the index (Section 3). For example, here we name the final assembler in Figure 5 as node ABCD, instead of node $n$.

The procedure of finding the optimal assembly supply chain is divided into three steps: **Step I**: Enumerate all possible supply chain candidates that have one final assembler in the last echelon and four suppliers in the most upstream echelon, providing module A, B, C and...
Figure 6: There are five configurations to connect the four nodes in the most upstream echelon to the final assembler D respectively.

**Step II**: For each supply chain candidate, based on the demand vector of the final assembler, \( q_{ABCD} = (d_1, d_2, \ldots, d_{12}) \), calculate the demand vector of all other nodes in the supply chain by Equation (2) and substitute them back to Equation (6) to get the supply chain complexity.

**Step III**: Compare the complexity value of all supply chain candidates and obtain the optimal assembly supply chain by selecting the one with the minimum complexity.

Among these three steps, the first step, enumerating all possible supply chain candidates, is the most challenging. It is because for a given number of nodes in the most upstream echelon, there are many ways to connect these nodes to the final assembler, which results in many different supply chain configurations. In addition, for each configuration there are more than one supply chain candidate due to different locations of the nodes in the most upstream echelon. For instance, in our example here, there are five different supply chain configurations, all of which have four nodes in the most upstream echelon, as shown in Figure 6. For each configuration, there are several possible supply chain candidates. For example, the configuration IV in Figure 6 has three different supply chains due to the location difference of the nodes in the most upstream echelon, as shown in Figure 7.

In order to enumerate all possible supply chain candidates, considering the configuration
Figure 7: There are more than one supply chain candidates for the same configuration difference and node location difference, a description of an assembly supply chain containing only characters of letters, ‘(‘ and ‘)’ is proposed here. Webbink and Hu [24] applied a similar string description to represent the system configurations in the context of manufacturing system design and developed a decomposition algorithm to generate manufacturing system configurations consisted of \( n \) workstations based on that description. In this paper, we used the similar description to represent assembly supply chains. In the description, one pair of parentheses represents an assembly relationship, including the assembly relationships between the nodes in the most upstream echelon and the virtual supplier. Starting from the most upstream echelon of the supply chain and moving forward to the final assembler, when an assembly relationship is met, a pair of parentheses is added. Repeat this process until the final assembler is reached. For instance, in Figure 7 (a), starting from the most upstream echelon, there are four supply-assembly relationships, between node A, B, C, D and the virtual supplier, which are represent as (A), (B), (C) and (D). Moving forward, module (A) and (C) are assembled together at node AC and one more parenthesis is added, through which sub-assembly ((A)(C)) is obtained. Sub-assembly ((B)(D)) is obtained in the similar way. Moving further, sub-assembly ((A)(C)) and ((B)(D)) are assembled at the final assembler and one more parenthesis is added. Therefore (((A)(C))((B)(D))) is obtained to
represent the supply chain of Figure 7 (a).

After developing this description method for assembly supply chains, we then develop an iterative decomposition algorithm, that generates all possible assembly supply chain candidates, given the number of nodes in the most upstream echelon. It is composed of the following three steps.

**Step 1:** According to the modules in the final product, generate the set of all sub-assemblies and modules, which could be produced by any node in all possible assembly supply chains. In the example of Figure 5, based on the four modules in the final product, A, B, C and D, the following set of sub-assemblies and modules is generated, (ABC), (ABD), (ACD), (BCD), (AB), (AC), (AD), (BC), (BD), (CD), (A), (B), (C), (D).

**Step 2:** List all the possible assembly combinations of sub-assemblies and modules generated from Step 1, through which the final product ABCD can be achieved and then one more parenthesis is added. For the example of Figure 5, the combinations are ((ABC)(D)), ((ABD)(C)), ((ACD)(B)), ((BCD)(A)), ((AB)(CD)), ((AB)(C)(D)), ((AC)(BD)), ((AC)(B)(D)), ((AD)(BC)), ((AD)(B)(C)), ((BC)(A)(D)), ((BD)(A)(C)), ((CD)(A)(B)) and ((A)(B)(C)(D)).

**Step 3:** For each assembly combination in step 2, check whether the cardinality of each inner parentheses is one or not. If the cardinality of an inner parentheses is more than one, it means there is a sub-assembly in this inner parenthesis. Then that sub-assembly is treated as the final product in Step 1. Go to step 1 and step 2. Repeat this process until the cardinalities of all inner parentheses are one, which means no more sub-assembly needs to be decomposed. For the combination instance of ((ABC)(D)), the cardinality of inner parentheses (ABC) is three, more than one. Treat sub-assembly (ABC) as the final product. Redo step 1 and Step 2, in which we get the following set, (AB), (AC), (BC), (A), (B), (C)
Figure 8: Iterative decomposition algorithm to generate all possible supply chain candidates and × stands for the infeasible candidates, which will be discussed in section 5.2 and possible assembly combinations, ((AB)(C)), ((AC)(B)), ((BC)(A)), ((A)(B)(C)). Take (((AB)(C))) for an instance and the cardinality of (AB) is more than one. Repeat Step 1 and Step 2. Only one assembly combination ((A)(B)) is obtained. Since the cardinality of all inner parentheses in ((A)(B)) is one, we stop. Replace (AB) in combination ((AB)(C)) with ((A)(B)) and then we get (((A)(B))(C)). Following the same procedure, we replace (ABC) in combination ((ABC)(D)) with (((A)(B))(C)) and finally obtain ((((A)(B))(C))(D)).

By this iterative decomposition algorithm, we can generate all possible supply chain candidates, shown in Figure 8. For each candidate, based on the demand vector of the final assembler, the demand vector of all the nodes in the supply chain is calculated by Equation (2) and then the corresponding complexity of that assembly supply chain is calculated through Equation (6). The optimal assembly supply chain is achieved through comparing complexity of all supply chain candidates and selecting the one with the least complexity.
5.2 Optimal Assembly Supply Chain Selection with Assembly Constraints

In the previous section, we developed an iterative algorithm to generate all possible supply chain candidates if the number of variants offered by the final assembler and their mix ratios are given. In that algorithm, we assume that there is no assembly sequence constraint and the final product can be obtained through any possible assembly sequence. But in practice, it may not be true. Assembly sequence constraints often exist in the product assembly process. For example, a laptop assembly process requires the keyboard to be assembled after all other components are assembled to the main board. In this section, we investigate the optimal assembly supply chain with assembly sequence constraints, given the number of variants at the final assembler and their mix ratios. The same example, shown in Figure 5, is used again to illustrate the procedure. Besides the knowledge of 12 variants offered by the final assembler with the demand share, $d_j, j = 1, \ldots, 12$, we also have the following assembly sequence constraints, represented in Figure 9: 1) Module A and B must be assembled before module C; 2) Module D can not be assembled until module C is assembled. If $i \succ j$ is used to represent a constraint that module $i$ must be assembled before module $j$, then the above assembly sequence constraints can be written as:

$$A \succ C, B \succ C, C \succ D$$

The procedure of finding the optimal supply chain with assembly sequence constraints is to first generate all feasible supply chain candidates that satisfy all the assembly sequence constraints and then compare complexity of these feasible supply chain candidates and finally obtain the optimal supply chain by selecting the one with the least complexity value. One
straightforward method of obtaining feasible supply chain candidates is to first generate all the supply chain candidates without assembly constraints by the iterative decomposition algorithm developed in section 5.1, then check the feasibility of each candidate and delete the candidates that do not satisfy the assembly constraints. This method can be illustrated in Figure 8, where × stands for the infeasible supply chain candidates that do not satisfy constraints (7).

This method is easy to understand and apply, but the efficiency is low because of the number of supply chain candidates generated under the assumption of no assembly sequence constraint, which increases exponentially with the increase of the number of modules in the product. Here, we develop a more efficient method, which generates all feasible supply chain candidates without generating and checking all possible supply chain candidates. Recall that in the iterative decomposition algorithm of section 5.1, we keep decomposing the assembly combinations obtained in step 2 until the cardinality of each inner parenthesis is one. Now, instead of checking the feasibility after all the decomposition is done, we check the feasibility of every intermediate supply chain candidate that is obtained after each decomposition. The infeasible intermediate supply chain candidates are deleted and in next decomposition iteration, no more decomposition is performed on these infeasible intermediate candidates, as shown in Figure 10. Compared with the previous method, the number of feasibility check is reduced, because infeasible intermediate supply chain candidates stop further decomposition.
Figure 10: Feasibility check of intermediate supply chain candidates after each decomposition and no more feasibility check is performed from this point.

Here the example shown in Figure 5, is used to illustrate this method. In this example, we want to generate all feasible supply chain candidates that satisfy the following assembly sequence constraints (7). First, in step 2 of the iterative decomposition algorithm, we obtain the following 14 intermediate supply chain candidates after 1st decomposition, ((ABC)(D)), ((ABD)(C)), ((ACD)(B)), ((BCD)(A)), ((AB)(CD)), ((AB)(C)(D)), ((AC)(BD)), ((AC)(B)(D)), ((AD)(BC)), ((AD)(B)(C)), ((BC)(A)(D)), ((BD)(A)(C)), ((CD)(A)(B)) and ((A)(B)(C)(D)). Second, we check the feasibility of these 14 intermediate supply chain candidates before we start the next decomposition. The intermediate supply chains that do not satisfy the constraints, are deleted. Third, the 2nd decomposition only starts with the following three feasible ones, ((ABC)(D)), ((AB)(C)(D)) and ((A)(B)(C)(D)). Then the same procedure repeats until the last decomposition is done. Figure 10 summaries the details of this method. This method helps to reduce the number of feasibility check from 26
In the iterative decomposition algorithm developed in section 5.1, we start from the final product and in each iteration, we decompose the final product (1st iteration) or the sub-assemblies (remaining iterations) into assembly combinations, which are composed of smaller-size sub-assemblies and modules. Let \( P_k \) represents the sub-assembly, which is decomposed in the \( k^{th} \) iteration (notice that \( P_1 \) is the final product in the first iteration). Here we use the following criteria to check whether constraint \( i \succ j \) is satisfied in one intermediate assembly combination, which is generated through the decomposition of sub-assembly \( P_k \).

First, we check whether sub-assembly \( P_k \) contains both module \( i \) and module \( j \). If not, we conclude that this assembly combination satisfies constraint \( i \succ j \), because the feasibility has already been guaranteed in the previous decompositions. If \( P_k \) contains both module \( i \) and \( j \), then we check whether this assembly combination, generated through the decomposition of \( P_k \), satisfies one of the following two requirements: 1) In this decomposition, \( P_k \) is decomposed into the combination of module \( j \) and other sub-assemblies (or modules); 2) In this decomposition, \( P_k \) is decomposed into the combination of a sub-assembly containing module \( j \) and other sub-assemblies (or modules) and this sub-assembly containing module \( j \) also contains module \( i \). If one of the above two requirements is met, then it is concluded that constraint \( i \succ j \) is satisfied in this intermediate assembly combination; otherwise, we conclude that constraint \( i \succ j \) is not satisfied and this intermediate assembly combination is infeasible. It is deleted and no more decomposition is performed on this combination in the next iteration. Figure 11 summaries the details of this feasibility check criteria.

We use Figure 10 as an example to illustrate this procedure. In first iteration, \( P_1 = (ABCD) \) is decomposed into 14 intermediate assembly combinations. We choose one in-
Figure 11: Feasibility check criteria for an intermediate assembly supply chain

termediate combination, \(((ABC)(D))\), to check whether it satisfies constraints (7). For constraint \(A \succ C\), first we find that \(P^1 = (ABCD)\) contains both module \(A\) and module \(C\). Then the intermediate combination \(((ABC)(D))\) is obtained through decomposing \(P^1 = (ABCD)\) into a sub-assembly \((ABC)\) that contains module \(C\) and another module \((D)\). The sub-assembly containing module \(C\), \((ABC)\), also contains module \(A\). So we conclude that assembly combination \(((ABC)(D))\) satisfies constraint \(A \succ C\). Similarly, intermediate combination \(((ABC)(D))\) also satisfies constraint \(B \succ C\). For constraint \(C \succ D\), the intermediate combination \(((ABC)(D))\) is obtained through decomposing \(P^1 = (ABCD)\) into module \(D\) and another sub-assembly \((ABC)\), so constraint \(C \succ D\) is also met. Then we conclude that \(((ABC)(D))\) is a feasible intermediate supply chain candidate and in next iteration, it will be further decomposed. Then in the second iteration, \(i = 2\), the sub-assembly \((ABC)\), i.e., \(P^2 = (ABC)\), is decomposed into the following four intermediate assembly combinations \(((AB)(C))\), \(((AC)(B))\), \(((BC)(A))\) and \(((A)(B)(C))\). Here we select combination \(((AC)(B))\) as an example to check whether it is a feasible intermediate combination. For constraint \(C \succ D\), the sub-assembly decomposed in this iteration, \(P^2 = (ABC)\), does not contain module \(D\). Therefore, we conclude that \(((AC)(B))\) satisfies constraint \(C \succ D\). For constraint \(B \succ C\), first the sub-assembly decomposed \(P^2 = (ABC)\) contains both module
B and module C. Then in this decomposition, the intermediate sub-assembly ((AC)(B)) is obtained through decomposing $P^2 = (ABC)$ into a sub-assembly (AC) that contains module C and another module (B). The sub-assembly containing module C, (AC), does not contain module B. Therefore, intermediate assembly combination ((AC)(B)) does not satisfy constraint $A \succ C$ and is an infeasible intermediate supply chain. So we delete it and in next iteration, no further decomposition will performed.

6 Discussions

One challenge of this complexity research is that when the iterative decomposition algorithm is used to generate possible supply chain candidates, the number of candidates increases exponentially as the number of modules in the product increases. If the number of modules is big, it is difficult to generate all possible supply chain candidates, especially when no assembly constraint exists. As discussed in section 3, if the demand shares of the variants at the final assembler fall into one of the following two extreme scenarios, one dominant variant and equal demand shares, the optimal supply chain can be easily obtained without generating all possible supply chain candidates. But unfortunately, very often the final assembler’s demand does not fit into these two scenarios and the iterative decomposition algorithm has to be used. So in these cases, it is very challenging to study the optimal supply chain problem if the number of modules is high and there is no assembly sequence constraint. Reformulating the problem by some mathematical programming tools, such as mixed integer programming (MIP) and dynamic programming (DP), probably could provide some alternative solution. For the particular problem of finding the optimal supply chain,
DP is more promising because most times the optimization problems formulated by MIP are NP-hard. However, the following two factors could be the challenges if we want to use DP to solve the problem. First, how the problem is formulated will definitely influence the solution efficiency. For example, based on Figure 8, the optimal supply chain problem can be formulated into the shortest path problem and DP can be used to find the solution. But in that case, the number of nodes still increases exponentially when the number of modules increases and therefore the computation burden does not reduce in this way. Therefore, how to find a better way to formulate the problem so that the number of states will not increase exponentially is critical for this optimization problem. Second, no matter how we formulate the problem, finally it will be a complexity minimization problem and in each stage we need to calculate the complexity incremental value based on the decision made at that state. According to the assumption of DP, this complexity incremental value should be independent from the future decisions. But recall that in our complexity definition, shown in Equation (6), the complexity value of each node is related to the total number of arcs in the supply chain, $K$, which is determined by the decision of current state and all future states. So in our problem, this assumption is violated. However, we notice that $2m \leq K \leq 2m + (m - 2)$, where $m$ is the number of nodes in the most upstream echelon. The lower bound, $K = 2m$, is the case of non-modular supply chain where the number of arcs in the supply chain is minimum, while the upper bound $K = 2m + (m - 2)$, is the case of modular supply chain with largest number of sub-assemblers, $m - 2$, where the number of arcs in the supply chain is maximum. This relationship could help us to apply DP in solving the problem by setting $K$ to the upper or lower bound.
7 Conclusions

In this paper, a complexity measure of assembly supply chains is derived based on Shannon’s information entropy. This complexity measure incorporates the detailed information of the supply chain structure, the number of variants offered by each node in the supply chain, and the mix ratios of the variants at each node. It is applied to solving the problem of finding the optimal assembly supply chain given the number of variants offered at the final assembler and the mix ratios of these variants. The optimal assembly supply chain configuration is theoretically studied in the following two special scenarios: 1) there is one dominant variant among all the variants offered by the final assembler, and 2) demand share are equal across all variants at the final assembler. It is shown that in the scenario of one dominant variant, the optimal assembly supply chain should be non-modular; but in the scenario of equal demand shares, the modular supply chain is more beneficial than non-modular when the product variety is high. For the general demand at the final assembler, a methodology is developed to find the optimal supply chain if no assembly sequence constraint exists. An iterative algorithm is proposed to generate all supply chain candidates. Then the methodology is extended to solve the optimal supply chain problem with the existence of assembly sequence constraints. A efficient method is developed to obtain all feasible supply chain candidates, satisfying all the assembly constraints. The research of supply chain complexity generates new insights on the influence of product variety on supply chains performance in mass customization. It provides a model based analytic method, instead of empirical case study, to the supply chain configuration selection problem in the presence of product variety. The model and algorithms developed in this paper can assist in making decisions such as when
and how to implement a modular assembly supply chain and how much variety should be economically offered.

**Acknowledge**

The authors gratefully acknowledge the financial support from the National Science Foundation Engineering Research Center for Reconfigurable Manufacturing Systems (Award No. EEC-9529125) and GOALI Project with General Motors (Award number: CMMI-0825438), both at the University of Michigan.

**References**


in a multi-tier system with modular assembly,” Management Science, 50(9), pp. 1293–
1308.

York.

Sons, 2 ed.


essment of hepatitis C virus sequence complexity by electrophoretic mobilities of both
single- and double-stranded DNAs,” Journal of Clinical Microbiology, 26(10), pp. 2982–
2989.

information distances, and complexity in atoms,” Journal of Chemical Physics, 123(17),
p. 17411.

645–655.


Appendix: Proofs

Proof of Proposition 1

Proof of (a)

The demand vector of the final assembler, node $n$, is $\mathbf{q}_n := (q_{n1}, q_{n2}, \ldots, q_{n,V_n})$. For notation convenience, we assume the dominant variant is variant 1 and then the demand share of the dominant variant is $q_{n1}$.
For node \( i = 1, 2, \ldots, n - 1 \), we can divide the variants offered by the final assembler into \( V_i \) disjoint subsets, each containing \( V_n/V_i \) variants of the final assembler, and the demand share of the variants in each subset add up to the demand share of a variant at node \( i \). Without loss of generality, here we assume the demand share of variant 1 offered by node \( i \) is obtained through summing the subset of variants, containing variant 1 from the final assembler. Therefore, when the demand share of variant 1 at the final assembler increases and approaches to 1, i.e., \( q_n \rightarrow 1 \), the demand share of variant 1 at node \( i = 1, 2, \ldots, n - 1 \) also increases and approaches to 1, i.e., \( q_i \rightarrow 1 \). Because \( \sum_{j=1}^{V_i} q_{ij} = 1 \) and \( q_{i1} \rightarrow 1 \) for \( i = 1, \ldots, n \), then the demand share of other variants at node \( i \) must approach 0, i.e., \( q_{ij} \rightarrow 0, j = 2, \ldots, V_i \).

The complexity of an assembly supply chain defined in Equation (6) is

\[
C = -\sum_{i=1}^{n} L_i \sum_{v=1}^{V_i} q_{iv} \log_2 \frac{q_{iv}}{K},
\]

where \( L_i \) is the number of supplies of node \( i \) and \( K \) is the number of arcs in the supply chain, i.e., \( K = \sum_{i=1}^{n} L_i \). With some algebra, it can be checked that the complexity of Equation (6) can also be written as

\[
C = \log_2 K - \frac{1}{K} \sum_{i=1}^{n} L_i \sum_{v=1}^{V_i} q_{iv} \log_2 q_{iv}.
\]

In the scenario of a dominant variant, we can rewrite the above equation as follows, where \( q_{i1} \rightarrow 1 \) and \( q_{ij} \rightarrow 0, j = 2, \ldots, V_i \)

\[
C = \log_2 K - \frac{1}{K} \sum_{i=1}^{n} L_i (q_{i1} \log_2 q_{i1} + \sum_{v=2}^{V_i} q_{iv} \log_2 q_{iv}).
\]

As \( q_{i1} \rightarrow 1 \), we have \( q_{i1} \log_2 q_{i1} = 1 \cdot \log_2 1 = 0 \). But as \( q_{ij} \rightarrow 0, j = 2, \ldots, V_i \), we have
\( q_{iv} \cdot \log_2 q_{iv} \), which is \( 0 \cdot \infty \) type limit and can be calculate through l’Hôpital’s rule,

\[
\lim_{q_{iv} \to 0} q_{iv} \log_2 q_{iv} = \lim_{q_{iv} \to 0} \frac{q_{iv} \log_2 q_{iv}}{1/q_{iv}} = \lim_{q_{iv} \to 0} \frac{(q_{iv} \log_2 q_{iv})'}{(1/q_{iv})'} = \lim_{q_{iv} \to 0} -\frac{1}{q_{iv}^2} = 0.
\]

Then in the scenario of one dominant variant, the complexity of an assembly chain, represented as Equation (9), is \( C = \log_2 K \), where \( K \) is the number of total arcs in the supply chain.

**Proof of (b)**

We want to make the decision of the supply chain configuration, given the number of variants offered by the final assembler and the mix ratios of these variants. Following proof (a), in the scenario of one dominant variant, the complexity of an assembly supply chain is \( C = \log_2 K \), where \( K \) is the number of total arcs in the supply chain, including the arcs from the virtual suppliers to the nodes in the most upstream echelon. Then the optimal assembly supply chain should have the minimum number of arcs in the supply chain. According to the definition of assembly supply chains, every node, except the virtual supplier and the final assembler, only has one downstream node, then we can obtain the following relationship, where \( n \) is the number of nodes in the supply chain and \( m \) is the number of nodes in the most upstream echelon,

\[
K = (n - 1) + m \quad (10)
\]

Since \( m \) is fixed by the number of module in the final product, in order to obtain the minimum value of \( K \) in Equation (10), we should choose the assembly supply chain with minimum number of nodes in the supply chain, which is the non-modular assembly supply chain. So
under the scenario of one dominant variant, the optimal assembly supply chain configuration is non-modular.

Proof of Proposition 2

Suppose the demand shares are equal across all the variants provided by the final assembler, node \( n \), i.e., \( q_n = (\frac{1}{V_n}, \ldots, \frac{1}{V_n})_{1 \times V_n} \). By Equation (2), it is easily to verify that the demand shares are also identical across all the variants at other nodes in the supply chain. In addition, here we assume all the nodes in the most upstream echelon, node 1, \ldots, m, provide the same number of variants, i.e, \( V_1 = \ldots = V_m = V \). So the demand vector of the nodes in the most upstream echelon takes the form of \( q_i = (\frac{1}{V}, \ldots, \frac{1}{V})_{1 \times V}, i = 1, \ldots m. \) Since we assume that a node can assemble any combination of the components from its upstream suppliers, then \( V_n = \prod_{i=1}^{m} V_i = V^m \).

We want to make the decision of the supply chain configuration, given the number of variants offered by the final assembler and the mix ratios of these variants. All the supply chains, which provide the same number of variants at the final assembler and have the same mix ratios of these variants, must have the same number of nodes in the most upstream echelon, \( m \). Next, we will prove that when \( V \) is big enough, there is at least one modular assembly supply chain, which has less complexity value than non-modular assembly supply chain.

Let us consider the following two assembly supply chains, which has the same demand vector of the final assembler, \( q_n = (\frac{1}{V_n}, \ldots, \frac{1}{V_n})_{1 \times V_n} = (\frac{1}{V^m}, \ldots, \frac{1}{V^m})_{1 \times V^m} \). One is non-modular assembly supply chain in Figure 12(a) and another is modular assembly supply chain
Figure 12: One non-modular assembly supply chain and one modular assembly with only one intermediate sub-assembler in Figure 12(b), which has one sub-assembler in the middle echelon, assembling components from node 1, \ldots, m-1. The demand vector of this sub-assembler in modular assembly supply chain is \( q_{n-1} = (\frac{1}{V_{m-1}}, \ldots, \frac{1}{V_{m-1}})_{1 \times V^{m-1}} \). The complexity of an assembly supply chain can be calculated through Equation (8), i.e., \( C = \log_2 K - \frac{1}{K} \sum_{i=1}^{n} L_i \sum_{v=1}^{V_i} q_{iv} \log_2 q_{iv} \).

For non-modular assembly supply chain, we substitute \( q_i = (\frac{1}{V}, \ldots, \frac{1}{V})_{1 \times V}, L_i = 1, i = 1, \ldots, m, q_n = (\frac{1}{V_m}, \ldots, \frac{1}{V_m})_{1 \times V_m}, L_n = m \) and \( K = 2m \) back to Equation (8) and get the complexity of the non-modular assembly supply chain in Figure 12(a),

\[
C_{\text{non-mod}} = \log_2 2m + \frac{m + 1}{2} \log_2 V
\]

Following the same argument, by substituting \( q_i = (\frac{1}{V}, \ldots, \frac{1}{V})_{1 \times V}, L_i = 1, i = 1, \ldots, m, q_{n-1} = (\frac{1}{V_{m-1}}, \ldots, \frac{1}{V_{m-1}})_{1 \times V^{m-1}}, L_{n-1} = m - 1, q_n = (\frac{1}{V_m}, \ldots, \frac{1}{V_m})_{1 \times V_m}, L_n = 2 \) and \( K = 2m + 1 \) back to Equation (8), we can get the complexity of the modular assembly supply chain in Figure 12(b),

\[
C_{\text{mod}} = \log_2(2m + 1) + \frac{m^2 + m + 1}{2m + 1} \log_2 V
\]

Then the complexity difference between modular and non-modular assembly supply chain in Figure 12 is,

\[
C_{\text{mod}} - C_{\text{non-mod}} = \log_2 \frac{2m + 1}{2m} - \frac{m - 1}{2(2m + 1)} \log_2 V \tag{11}
\]
Since \( m \) is the number of nodes in the most upstream echelon of the assembly supply chain, \( m \) must be an integer greater than 1, i.e., \( m > 1, a \in Z \), which results in \( \frac{2m+1}{2m} > 1 \) and \( \frac{m-1}{2(2m+1)} > 0 \). For a fixed \( m \), define the following two constants, \( A = \log_2 \frac{2m+1}{2m} > 0 \) and \( B = \frac{m-1}{2(2m+1)} > 0 \). Then the difference between non-modular and modular assembly supply chain in Equation (11), can be rewritten as \( C_{\text{mod}} - C_{\text{non-mod}} = A - B \log_2 V \), which is a decreasing function of \( V \). There must be a threshold, \( t \), so that when \( V \geq t \), \( C_{\text{mod}} - C_{\text{non-mod}} < 0 \), which makes the modular assembly chain more preferable. So when \( V \) is big enough, i.e., \( V \geq t \), we find at least one modular assembly supply chain, shown in Figure 12(b), has less complexity value than the corresponding non-modular assembly supply chain, shown in Figure 12(a).

Proof of Proposition 3

Suppose the demand shares are equal across all the variants provided by the final assembler, node \( n \), i.e., \( q_n = (\frac{1}{V_n}, \ldots, \frac{1}{V_n})_{1 \times V_n} \). By Equation (2), it is easily to see that the demand shares are also identical across all the variants at other nodes in the supply chain. In addition, since all the nodes in the most upstream echelon, node 1, \ldots, \( m \), provide the same number of variants, i.e, \( V_1 = \ldots = V_m = V \), then we have \( q_i = (\frac{1}{V}, \ldots, \frac{1}{V})_{1 \times V}, i = 1, \ldots m \) and \( V_n = \prod_{i=1}^{m} V_i = V^m \).

Proof of (a)

From the proposition statement, it is easy to see that \( m \), can be written as the form of the product of two positive integers, i.e., \( m = c \times d, c, d \in Z^+ \). Now consider the following
two supply chains: Both of them satisfy the above assumptions and have one intermediate echelon, in which there are several sub-assemblers. Supply chain 1 has \( c \) intermediate sub-assemblers, each of which has \( d \) suppliers from the most upstream echelon and then has the demand vector as \( q_i^{(1)} = \left( \frac{1}{V^a}, \ldots, \frac{1}{V^a} \right)_{1 \times V^a}, i = m + 1, \ldots, m + c \). Supply chain 2 has \( d \) intermediate sub-assemblers, each of which has \( c \) suppliers from the most upstream echelon and then has \( q_i^{(2)} = \left( \frac{1}{V^d}, \ldots, \frac{1}{V^d} \right)_{1 \times V^d}, i = m + 1, \ldots, m + d \).

Recall in the proof of Proposition 1, the complexity of an assembly supply chain can be calculated through Equation (8), i.e., \( C = \log_2 K - \frac{1}{R} \sum_{i=1}^{n} L_i \sum_{v=1}^{V_i} q_{iv} \log_2 q_{iv} \). For supply chain 1, there are total \( 2m + c \) arcs in the supply chain, i.e., \( K^{(1)} = 2m + c \). We substitute \( q_i^{(1)} = \left( \frac{1}{V^a}, \ldots, \frac{1}{V^a} \right)_{1 \times V^a}, L_i^{(1)} = 1, i = 1, \ldots, m \) and \( q_i^{(1)} = \left( \frac{1}{V^d}, \ldots, \frac{1}{V^d} \right)_{1 \times V^d}, L_i^{(1)} = d, i = m + 1, \ldots, m + c \) and \( q_n^{(1)} = \left( \frac{1}{V^m}, \ldots, \frac{1}{V^m} \right)_{1 \times V^m}, L_n^{(1)} = c \) and \( K^{(1)} = 2m + c \) back to Equation (8) to get the complexity of the supply chain 1,

\[
C^{(1)} = \log_2 (2m + c) + \frac{m + md + mc}{2m + c} \log_2 V
\]

Similarly, for supply chain 2, there are total \( 2m + d \) arcs in the supply chain, i.e., \( K^{(2)} = 2m + d \). We substitute \( q_i^{(2)} = \left( \frac{1}{V^a}, \ldots, \frac{1}{V^a} \right)_{1 \times V^a}, L_i^{(2)} = 1, i = 1, \ldots, m \) and \( q_i^{(2)} = \left( \frac{1}{V^d}, \ldots, \frac{1}{V^d} \right)_{1 \times V^d}, L_i^{(2)} = c, i = m + 1, \ldots, m + d \) and \( q_n^{(2)} = \left( \frac{1}{V^m}, \ldots, \frac{1}{V^m} \right)_{1 \times V^m}, L_n^{(2)} = d \) and \( K^{(2)} = 2m + d \) back to Equation (8) to get the complexity of the supply chain 2,

\[
C^{(2)} = \log_2 (2m + d) + \frac{m + md + mc}{2m + d} \log_2 V
\]

Then the complexity difference between these two supply chains is,

\[
C^{(1)} - C^{(2)} = \log_2 \frac{2m + c}{2m + d} + \frac{m + md + mc}{(2m + c)(2m + d)} (d - c) \log_2 V
\]

Let \( A = \log_2 \frac{2m + c}{2m + d} \) and \( B = \frac{m + md + mc}{(2m + c)(2m + d)} (d - c) \). If \( c \leq d \), then \( A \leq 0 \) and \( B \geq 0 \) and we
can rewrite the complexity difference as $C^{(1)} - C^{(2)} = A + B \cdot \log_2 V$, which is an increasing function of $V$. Then there must be a threshold, $\eta$, so that when $V \geq \eta$, $C^{(1)} - C^{(2)} \geq 0$. We can conclude that when $c \leq d$ and $V \geq \eta$, $C^{(1)} \geq C^{(2)}$, which means the modular assembly supply chain with larger number of intermediate sub-assemblers has less complexity value.

Proof of (b)

Now consider the following two supply chains: Both of them have one intermediate echelon, where there are $c$ sub-assemblers. In supply chain 1, all intermediate sub-assemblers have the same number of suppliers from the most upstream echelon, $d$, i.e., $L^{(1)}_{m+i} = d, i = 1, \ldots, c$, and $\sum_{i=1}^{c} L^{(1)}_{m+i} = cd = m$. In supply chain 2, each intermediate sub-assembler has different number of suppliers from the most upstream echelon. But since one node can only supply one downstream node, then for supply chain 2, the sum of the number of suppliers for all these $c$ sub-assemblers is still $cd$, i.e., $\sum_{i=1}^{c} L^{(2)}_{m+i} = cd = m$. We can rewrite the number of suppliers for these $c$ sub-assemblers in supply chain 2 in the following form, $L^{(2)}_{m+i} = d + e_i, e_i \in \mathbb{Z}, \sum_{i=1}^{c} e_i = 0, i = 1, \ldots, c$.

Each sub-assembler in supply chain 1 has $d$ suppliers from the most upstream echelon, resulting the following demand vectors: $q^{(1)}_{m+i} = \left(\frac{1}{V^{d+e_i}}, \ldots, \frac{1}{V^{d+e_i}}\right)_{1 \times V^d}, i = 1, \ldots, c$. Each sub-assembler in supply chain 2 has $d + e_i$ suppliers from the most upstream echelon, which results in the following demand vector: $q^{(2)}_{m+i} = \left(\frac{1}{V^{d+e_i}}, \ldots, \frac{1}{V^{d+e_i}}\right)_{1 \times V^{d+e_i}}, i = 1, \ldots, c$.

The complexity of an assembly supply chain can be calculated through Equation (8), i.e., $C = \log_2 K - \frac{1}{R} \sum_{i=1}^{n} L_i \sum_{v=1}^{V_i} q_{iv} \log_2 q_{iv}$. We substitute $q^{(1)}_{i} = \left(\frac{1}{V}, \ldots, \frac{1}{V}\right)_{1 \times V}, L^{(1)}_{i} = 1, i = 1, \ldots, m$ and $q^{(1)}_{m+i} = \left(\frac{1}{V^{d+e_i}}, \ldots, \frac{1}{V^{d+e_i}}\right)_{1 \times V^d}, L^{(1)}_{m+i} = d, i = 1, \ldots, c$ and $q^{(1)}_{n} = \left(\frac{1}{V^m}, \ldots, \frac{1}{V^m}\right)_{1 \times V^m}, L^{(1)}_{n} =$.
and \( K^{(1)} = 2m + c \) back to Equation (8) to get the complexity of the supply chain 1,

\[
C^{(1)} = \log_2(2m + c) + \frac{m + cd^2 + mc}{2m + c} \log_2 V
\]

Similarly, we substitute \( q_i^{(2)} = (\frac{1}{V}, \ldots, \frac{1}{V})_{1 \times V}, L_i^{(2)} = 1, i = 1, \ldots, m \) and \( q_{m+i}^{(2)} = (\frac{1}{V^{d+e_i}}, \ldots, \frac{1}{V^{d+e_i}})_{1 \times V^{d+e_i}}, L_i^{(2)} = d + e_i, i = 1, \ldots, c \) and \( q_n^{(2)} = (\frac{1}{V^{m}}, \ldots, \frac{1}{V^{m}})_{1 \times V^{m}}, L_n^{(2)} = c \) and \( K^{(2)} = 2m + c \) back to Equation (8) to get the complexity of the supply chain 2,

\[
C^{(2)} = \log_2(2m + c) + \frac{m + c \sum_{i=1}^{c} (d + e_i)^2 + mc}{2m + c} \log_2 V
\]

The complexity difference between supply chain 1 and supply chain 2 is,

\[
C^{(1)} - C^{(2)} = \frac{c \log_2 V}{2m + c} \left[ d^2 - \sum_{i=1}^{c} (d + e_i)^2 \right]
\]

\[
= \frac{c \log_2 V}{2m + c} \left[ d^2 - \sum_{i=1}^{c} (d^2 + 2de_i + e_i^2) \right]
\]

\[
= -\frac{c \log_2 V}{2m + c} \left[ 2d \sum_{i=1}^{c} e_i + \sum_{i=1}^{c} e_i^2 \right]
\]

Recall that \( \sum e_i = 0 \), then \( C^{(1)} - C^{(2)} = -\frac{c \log_2 V}{2m + c} \sum_{i=1}^{c} e_i^2 \). Since \( V \geq 1, m, c \in \mathbb{Z}^+ \) and \( \sum_{i=1}^{c} e_i^2 \geq 0 \), we get \( C^{(1)} - C^{(2)} \leq 0 \). So it is concluded that in the scenario of equal demand shares, the complexity of supply chain 1 is less than supply chain 2.

**List of Figures**

1. Manufacturers move from non-modular assembly supply chains to modular assembly supply chains .................................................. 3
2. A general assembly supply chain and relationships of variants’ demand share 9
3. The example used to illustrate the calculation of assembly supply chain complexity .................................................. 13

Hui Wang MANU-09-110 45
Three modular assembly supply chains with 6 nodes in the most upstream echelon and one intermediate echelon ........................................... 19

The example used to illustrate the methodology of finding the optimal assembly supply chain .......................................................... 21

There are five configurations to connect the four nodes in the most upstream echelon to the final assembler ........................................... 22

There are more than one supply chain candidates for the same configuration 23

Iterative decomposition algorithm to generate all possible supply chain candidates ................................................................. 25

A set of assembly sequence constraints .................................................. 27

Feasibility check of intermediate supply chain candidates after each decomposition ........................................................... 28

Feasibility check criteria for an intermediate assembly supply chain .............. 30

One non-modular assembly supply chain and one modular assembly with only one intermediate sub-assembler ................................. 41