J Intell Manuf (2018) 29:259–275
https://doi.org/10.1007/s10845-015-1106-9

Modeling and measuring the structural complexity in assembly supply chain networks

Nima Hamta[1,2] · M. Akbarpour Shirazi[1] · Sara Behdad[2] · S.M.T. Fatemi Ghomi[1]

Received: 3 January 2015 / Accepted: 1 June 2015 / Published online: 23 June 2015

Abstract Complexity of assembly supply chains (ASCs) is a challenge for designers and managers, especially when ASC systems become increasingly complex due to technological developments and geographically various sourcing arrangements. One of the major challenges at the early design stage is to make decision about an appropriate configuration of ASC. This paper addresses modeling and measuring the structural complexity of ASC networks in order to establish a framework obtaining the optimal ASC configuration. Considering relationship between supply chains and assembly systems, structural complexity measures for ASC network and assembly lines inside the network are developed based on Shannon's information entropy. This complexity model can be used to configure supply chain networks and assembly systems with robust performance. In order to generate different feasible configurations of ASCs, a four-step algorithm is proposed considering assembly sequence constraint. Finally, the optimal ASC network is obtained by comparing the total complexity values of the feasible configurations.

Keywords Structural complexity · Assembly supply chain network · Assembly line · Optimal configuration

✉ M. Akbarpour Shirazi
akbarpour@aut.ac.ir

1 Department of Industrial Engineering, Amirkabir University of Technology (Tehran Polytechnic), 424 Hafez Avenue, 1591634311 Tehran, Iran

2 Department of Industrial and Systems Engineering, University at Buffalo, The State University of New York, 342 Bell Hall, Buffalo, NY 14260-2050, USA

Introduction

Assembly supply chain (ASC) as an important research area in supply chain management has attracted the research attention in the past decade (İnkaya and Akansel 2015). ASCs consist of several different entities in which assembly activities play a significant role to produce differentiated and/or undifferentiated products (Modrak and Marton 2012). Two types of ASC have been introduced in the related literature: traditional non-modular ASC in which all assembly activities are done by the final assembler, and modular ASC in which the final assembler allocates product modules to inter-mediate sub-assemblers and a few number of assembled modules will be transported to the final assembler. It is worth mentioning that modular assembly has been applied in many industries, for instance automotive and aerospace (Chiu and Okudan 2014; Hu et al. 2008). Figure 1 demonstrates the structure of these two ASC networks.

Since ASC systems are becoming increasingly complex because of technical advancements, complexity modeling of these systems is a new challenge for designers and managers. A complex system contains a lot of components, elements or agents which interact with each other and with the environment whereas there is inherently uncertainty throughout the design or development process, and the system's outcome would not be fully controllable or predictable (ElMaraghy et al. 2012). The understanding of complexity in production systems leads to develop methods for decreasing the degree of complexity and therefore designing effective and predictable systems. Modeling of complexity helps in designing systems with robust performances in terms of cost, time, quality and flexibility (Zeltzer et al. 2013). In addition, complexity modeling is an efficient way to find the methods in order to reduce the complexity. Reduction of redundant complexity of ASC can increase organizational performance and

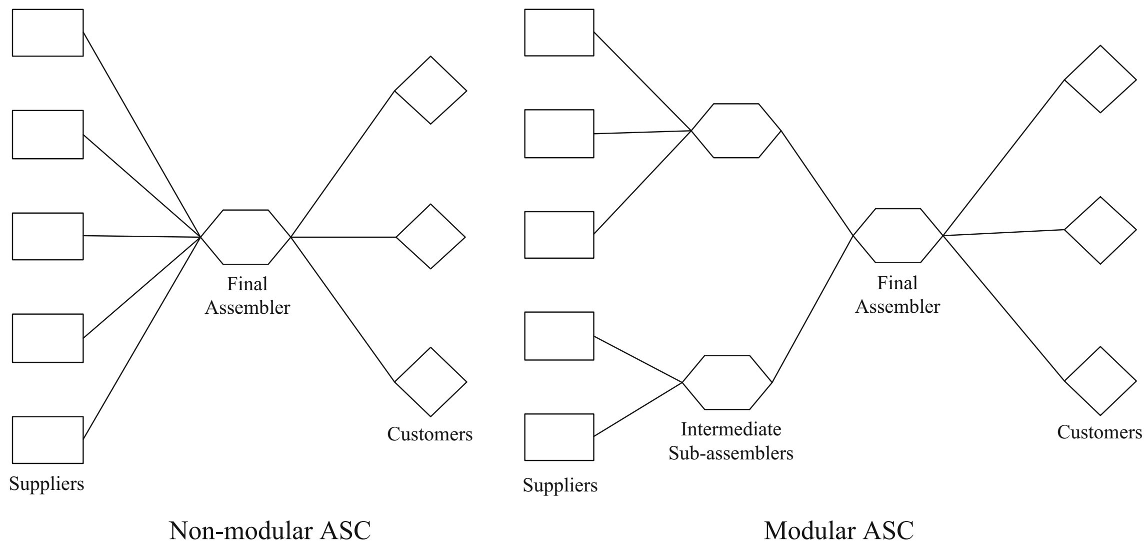


Fig. 1 The structure of non-modular and modular ASC networks

reduce operational inefficiencies (Modrak and Marton 2013). In general, high complexity of ASC systems makes them difficult to analyze, because a small change may lead to an enormous reaction.

The measure of complexity has been concerned in production/manufacturing systems in different ways. However, there are a few studies that define complexity in an obvious manner. In an effort to understand complexity, Schuh et al. (2008) define its main drivers as: uncertainty, dynamics, multiplicity, variety, interactions and interdependencies and combination of these properties can determine whether a considered system is complex or not. Recently, ElMaraghy et al. (2012) have provided a comprehensive review on complexity models in engineering design, manufacturing and business. They believe that designing systems with less complexity are main issues for further research.

From a basic point of view, two types of complexity are defined in the related literature: static (structural) complexity which deals with the configuration and structure of a system's components, defined as the expected amount of information required to describe the state of a system, whereas dynamic (operational) complexity addresses the behavior of the system over time and the probability of the system to be in control. Dynamic complexity is the expected amount of information required to describe the state of a system deviating from its design performance because of uncertainty (Frizelle and Woodcock 1995; Papakostas et al. 2009).

Two other well-known types of complexity have been proposed in the literature: time-independent and time-dependent (Suh 2005). The same as static complexity definition, time-independent complexity is a consequence of not satisfying the system's functional requirements at all times, including uncertainty that increase because of the designer's lack of understanding or knowledge about the system and its components. On the other hand, time-dependent complexity may be either a combined result of different complexities that increases as a function of time due to the continuous expansion of possible combinations of states over time (time-dependent combinatorial complexity), or periodic that exists in a finite time period with limited number of possible combinations of states as a smaller scale complexity (Wiendahl and Scholtissek 1994).

Therefore as it can be seen, static complexity is time-independent and deals with the product and system structure. The static complexity can be decreased by simplifying the design of systems and structure of products/processes (Webbink and Hu 2005). On the other hand, dynamic complexity is time-dependent and considers the operational behavior of the system.

There are different approaches in the literature to describe system complexity in the engineering design and manufacturing area [see for example (ElMaraghy et al. 2012; Suh 2005)]. The first approach is based on Shannon's information theory/entropy in which complexity defines as an entropy function of product variety (Hu et al. 2008). Frizelle and Woodcock (1995) propose a method using entropy to measure complexity in the structural and operational domains in manufacturing. The second complexity approach is based on the second axiom of axiomatic design theory that uses the information content as a measure of complexity in which information is utilized as a measure of uncertainty in achieving the functional requirements (Suh 1999). Uncertainty complexity is often evaluated via probability theory and formalized in the context of Shannon's entropy.

In the area of structural complexity of manufacturing systems, Blecker et al. (2005) mention that the structural complexity is caused by the static nature of processes, products, and structures, while the dynamic complexity arises from the external and internal sources within the operation, such as variations in times and amounts due to material shortage, insufficient supplier reliability or machine breakdowns. Deshmukh et al. (1998) and Sivadasan et al. (2006) describe that in the manufacturing systems and supply chains, structural complexity deals with schedule variety whereas operational complexity deals with deviating from the schedule due to uncertainty. Wilding (1998) defines a supply chain complexity triangle including deterministic chaos, demand amplification and parallel interactions to understand the generation of uncertainty within the supply chains. Milgate (2001) presents a conceptual model that identifies three basic dimensions of supply chain complexity and shows the linkage of uncertainty with delivery performance. He mentions there is no evidence that increased technological intricacy or more complicated organizational systems hamper performance.

In order to maintain or increase market share and avoid increasing costs, manufacturing organizations are using their current manufacturing system to produce customized products (Hamta et al. 2015). As a consequence, the large number of product variants significantly increases the complexity of manufacturing systems. In this regard, Zhu et al. (2008) propose a measure of manufacturing complexity introduced by product variety to quantify human performance in making various choices. They also develop models to evaluate the complexity at assembly workstations and its propagation through the assembly line. Wang et al. (2011) develop a multi-objective optimization approach for manufacturing complexity and variety in assembly systems through product variety selection. They introduce relative complexity to measure the complexity when designing a product family and the assembly system to find the best set of product variants.

This paper proposes a measure to calculate complexity of ASC networks considering the complexity inside assembly lines as a main stage of the supply chain. Although until now different approaches have been developed regarding manufacturing complexity to product and process structures and the human operator, to the best of our knowledge there is no paper in the related literature that measures the total structural complexity of ASC networks and assembly lines. In addition in this paper, a decomposition algorithm is proposed to generate all feasible ASC configurations while there is specific number of modules for producing a product family. Then the optimal ASC configuration would be selected among the generated feasible networks based on the minimum total complexity.

The rest of this paper is organized as follows. Section “Modeling and measuring of the complexity” presents the complexity modeling of the considered system in which measures for calculating the ASC network complexity and assembly line complexity are proposed. In Sect. “Illustrative numerical example”, a numerical example is given to illustrate the performance of the developed model. Section “Proposed algorithm to generate different ASC configurations” proposes an algorithm to generate different feasible configurations of ASCs. In sect. “Selection of optimal ASC network”, the optimal ASC configuration is selected. Finally, Sect. “Conclusions and future studies” is devoted to concluding remarks and some guidelines for future studies.

Modeling and measuring of the complexity

This section starts with a brief introduction to measure of complexity based on Shannon’s information theory. Then, the complexity measures of ASC network and assembly lines inside the supply chain are described. The complexity of the whole system is then obtained using these measures.

Measure of complexity

As already mentioned, different approaches have been developed in the literature to describe complexity measures and system complexity. One of the well-known approaches is based on Shannon’s information theory/entropy in which information is employed as a measure of uncertainty (Shannon 2001). Entropy is a measure of unpredictability in a random process. Shannon defines the entropy H of discrete random variable X with values $\{x_1, x_2, \ldots, x_n\}$ and probability mass function $P(X)$ as follows (Borda 2011):

$$H(X) = E\left[I(X)\right] = E\left[-\ln(P(X))\right], \tag{1}$$

where E is the expected value operator and I is the information content of X. When a finite sample is assumed, complexity as Shannon entropy can explicitly be formulated as follows:

$$H(X) = \sum_{i=1}^{n} P(x_i)I(x_i) = -c\sum_{i=1}^{n} P(x_i)\log P(x_i), \tag{2}$$

where c is a constant depending on the selected base of logarithm. Since $\log_2$ is commonly selected, $c = 1$ and the unit of complexity is bit. The unit of complexity is nat for Euler’s number e ($c = e$), and dit (or digit) for $c = 10$ (Zhu et al. 2008). In fact in Relation (2), H is the summation of surprisal functions weighted by probability Ps. A surprisal function $\log 1/P$ is defined to quantify how much uncertainty (surprise) is incurred for an individual process. As it is clear, the higher probability of the incoming alternative incurs the less surprisal and vice versa. Therefore, by weighting the surprisal with probabilities for the process, the entropy is obtained that characterizes the average randomness of a system. Hence, the entropy function H possesses most of the required properties to be a possible measure for complexity. It is worth mentioning that If Ps be closer to each other, H would increase and any change toward equalization of $P(x_i)$ would increase H. For a certain n when $P(x_i) = 1/n$, H is maximum and equal to log n (Shannon 2001).

ASC network complexity

In this subsection, the complexity model of ASC is presented based on Shannon's information entropy introduced in Sect. "Measure of complexity". In this regard, the detailed information on the supply chain structure, the number of variants each node (facility) produces and the mix ratios of the variants are employed. We assume the final product of the ASC is a product family with m different modules (also referred as functional features) as $M = \{M_1, M_2, \ldots, M_m\}$, where each module has several different variants (See Fig. 2) (Baud-Lavigne et al. 2014; Fujita et al. 2013). Let variants' set is $V_i = \{V_{i1}, V_{i2}, \ldots, V_{iOi}\}$, where V_{iv} denotes vth ($v = 1, 2, \ldots, O_i$) of module $i = 1, 2, \ldots, m + n$. Therefore, nodes $1, 2, \ldots, m$ of ASC are the most upstream echelon, i.e. the number of nodes in the most upstream echelon is equal to the number of modules in the product family (m). In addition, it is assumed that a downstream node can assemble any combination of the variants prepared by related upstream suppliers, and each combination is considered as a different variant. Since the nodes in the most upstream echelon do not have suppliers and we want to capture all the supply-assembly activities in the ASC network, a virtual supplier is considered, denoted as node 0, which supplies the raw materials of nodes in the most upstream echelon. After upstream suppliers, we assume there are $n - 1$ intermediate sub-assemblers to assemble any combination of the variants. Then, final assembler (node n) produces end-products. Finally, end-products are shipped to the customers based on their demands. Figure 3 shows the structure of considered ASC network.

The complexity of an ASC is caused by some important factors such as the supply chain structure determined by the number of nodes and links, demand uncertainty each node is confronted and product variety level of each node in the supply chain (Wang et al. 2010). It should be noted that the demand uncertainty a node confronts is specified by the mix ratios of the variants at that node, which

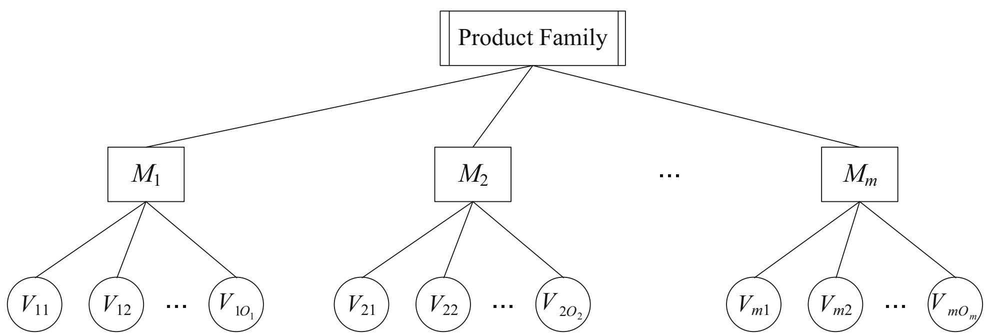


Fig. 2 The structure of a product family with different modules

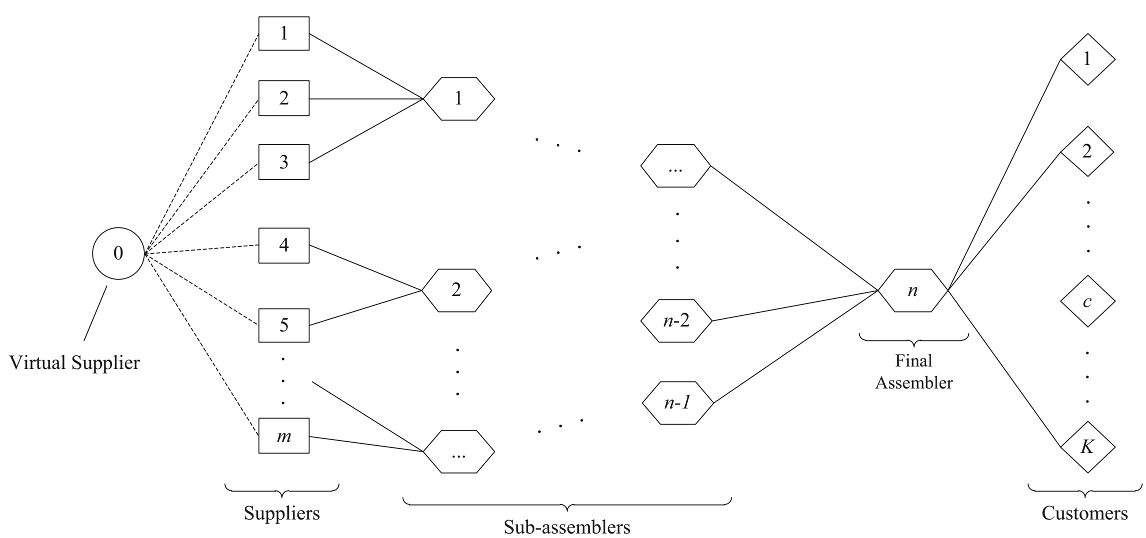


Fig. 3 The general structure of considered ASC network

expresses the probability of being a certain variant for the next required component. In this regard, p_{iv} is defined as the probably of demanding variant v at node i and vector $\boldsymbol{p}_i = (p_{i1}, p_{i2}, \ldots, p_{iOi})$ states the mix ratios of variants produced by node i.

Using information entropy concept presented in the previous subsection, the proposed complexity model of a general ASC is presented in the following steps:

Step 1- Adjacency matrix $\boldsymbol{\Delta}$ is defined in Relation (3) to represent the relationships of nodes in the ASC network where $\delta_{ij} = 1$ if node i is a supplier of node j $(i, j = 0, 1, \ldots, m+n)$; otherwise $\delta_{ij} = 0$.

$$\Delta = \begin{bmatrix} \delta_{00} & \delta_{01} & \cdots & \delta_{0,m+n} \\ \delta_{10} & \delta_{11} & \cdots & \delta_{0,m+n} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{m+n,0} & \delta_{m+n,1} & \cdots & \delta_{m+n,m+n} \end{bmatrix}_{(m+n+1)\times(m+n+1)} \quad (3)$$

Step 2- For every relationship with $\delta_{ij} = 1$ in adjacency matrix $\boldsymbol{\Delta}$, matrix $\boldsymbol{P}^{ij}$ is defined to represent the mix ratios of variants for nodes i and j. In matrix $\boldsymbol{P}^{ij}$, the number of rows is equal to the number of variants provided at node i, i.e. O_i, and the number of columns is the number of variants produced at node j, i.e. O_j.

$$P^{ij} = \begin{bmatrix} p_{1v}^{ij} \\ p_{2v}^{ij} \\ \vdots \\ p_{O_i,v}^{ij} \end{bmatrix} = \begin{bmatrix} p_{11}^{ij} & p_{12}^{ij} & \cdots & p_{1,O_j}^{ij} \\ p_{21}^{ij} & p_{22}^{ij} & \cdots & p_{2,O_j}^{ij} \\ \vdots & \vdots & \ddots & \vdots \\ p_{O_i,1}^{ij} & p_{O_i,2}^{ij} & \cdots & p_{O_i,O_j}^{ij} \end{bmatrix}_{O_i \times O_j} \quad (4)$$

In Relation (4), $p_{uv}^{ij} (u = 1, 2, \ldots, O_i, v = 1, 2, \ldots, O_j)$ is the production rate of variant u that node i produces for the production of variant v at node j to satisfy the customer's demand, assuming that the total demand for all variants at the final assembler is equal to 1.

Step 3- Using the following relation, every matrix $\boldsymbol{P}^{ij}$ is normalized where $\hat{\boldsymbol{P}}^{ij}$ is normalized $\boldsymbol{P}^{ij}$:

$$\hat{p}_{uv}^{ij} = p_{uv}^{ij} / \sum_{i=1}^{m+n} \sum_{j=1}^{m+n} \sum_{u=1}^{O_i} \sum_{v=1}^{O_j} p_{uv}^{ij} \quad (5)$$

Then, based on Shannon's information entropy introduced in Relation (2), we define the complexity measure of any relationship of the ASC network with $\delta_{ij} = 1$ in adjacency matrix $\boldsymbol{\Delta}$, as follows:

$$C_{ij} = -\sum_{u=1}^{O_i} \sum_{v=1}^{O_j} \hat{p}_{uv}^{ij} \log_2 \hat{p}_{uv}^{ij} \quad (6)$$

Step 4- Finally, the total complexity (C) of an ASC is calculated by summing the complexity measures of all relationships in the supply chain as follows:

$$C = \sum_{i=1}^{m+n} \sum_{j=1}^{m+n} C_{ij} \quad (7)$$

The complexity measure specifies the level of uncertainty in the flow of material occurring in the supply chain. By employing Relations (5) and (6), the complexity of an ASC can be computed via an easier and understandable formulation as follows, where $A = \sum_{i=1}^{m+n} \sum_{j=1}^{m+n} \sum_{u=1}^{O_i} \sum_{v=1}^{O_j} p_{uv}^{ij}$ is the total number of arcs in the ASC network:

$$\begin{aligned} C &= -\sum_{i=1}^{m+n} \sum_{j=1}^{m+n} \sum_{u=1}^{O_i} \sum_{v=1}^{O_j} \frac{p_{uv}^{ij}}{A} \log_2 \left(\frac{p_{uv}^{ij}}{A} \right) \\ &= -\frac{1}{A} \sum_{i=1}^{m+n} \sum_{j=1}^{m+n} \sum_{u=1}^{O_i} \sum_{v=1}^{O_j} p_{uv}^{ij} \left(\log_2 p_{uv}^{ij} - \log_2 A \right) \\ \rightarrow C &= \log_2 A - \frac{1}{A} \sum_{i=1}^{m+n} \sum_{j=1}^{m+n} \sum_{u=1}^{O_i} \sum_{v=1}^{O_j} p_{uv}^{ij} \log_2 p_{uv}^{ij} \end{aligned} \quad (8)$$

As it can be seen in Relation (8), the total number of arcs (A) has effect on the complexity measure according to the expectations. In addition, the complexity measure shows the amount of uncertainty about the next flow of materials in the ASC network.

Assembly line complexity

Assembly systems have been widely used for mass production especially in manufacturing customized products (Wang et al. 2011). In these systems, human operators are employed for assembly activities to handle the increasing variety of required products. Operators at each workstation of a manual assembly system must make correct selections of parts, tools, fixtures, and assembly procedures in a sequential manner (Su et al. 2012). These selections contribute to the complexity in the system. Examples of the corresponding selections are briefly expressed as follows (Zhu et al. 2008):

- Tool selection: pick up the correct tool according to the added part to be assembled in addition to the base part to be mounted on.
- Fixture selection: select the correct fixture according to the base part i.e., the partially completed assemblage to be mounted on in addition to the added part to be assembled.
- Procedure selection: select the correct procedure, such as approach angle, part orientation, or temporary unload of given parts due to geometric conflicts/subassembly stabilities.

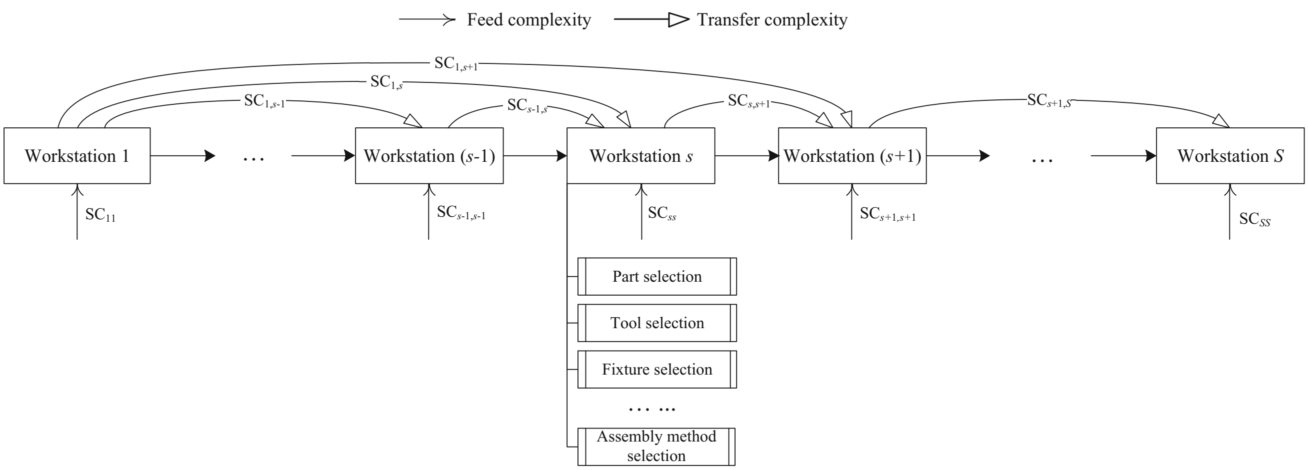


Fig. 4 Complexity propagation in an assembly line with selections at one workstation

Some assembly activities are caused by the variants added at the current workstation, such as picking up a part, or selection of tool(s) for a selected part. The complexity related to these activities is defined as feed complexity. On the other hand, selection of fixtures, tools or assembly procedures may depend on the variants added from an upstream workstation. This type of complexity is defined as transfer complexity (Hu et al. 2008). Figure 4 shows the propagation scheme of these two types of complexity in an assembly line while different selections in assembly activities at one workstation have been stated.

Therefore, the total complexity at a workstation in an assembler is simply obtained by summing the feed complexity at that workstation and the transfer complexity from the upstream workstation, i.e. the complexity SC_s for workstation s is as follows:

$$SC_s = SC_{ss} + \sum_{\forall r: r<s} SC_{rs} \tag{9}$$

where SC_{ss} (with two identical subscripts) is the feed complexity of workstation s, and SC_{rs} (with two different subscripts) is the complexity of workstation s caused by variants added at upstream workstation r. Suppose there is a set of assembly activities at each workstation in which each workstation has several selection alternatives. Let $A_s = \{A_{s1}, A_{s2}, \ldots, A_{sW}\}$ indicates the set of assembly activities at workstation s where A_{sw} denotes wth activity at workstation s. The selections required in the wth activity could be caused by the variety added at the current workstation (feed complexity), in addition to those of the upstream workstations (transfer complexity). If H_s^w denotes the entropy obtained from the variant mix ratio of wth activity at workstation s, the total complexity of workstation s would be weighted sum of the various types of selection complexity at that workstation as follows:

$$SC_s = \sum_{w=1}^{W} \lambda_s^w H_s^w \qquad \lambda_s^w > 0, \quad \sum_{w=1}^{W} \lambda_s^w = 1 \tag{10}$$

where λ_s^w indicates the weight related to the task difficulty of wth assembly activity at workstation s, depending on the nominal human performance. Now, assume matrix $\mathrm{Q} = ((q_{jv}))$ in which each element of the matrix is obtained using matrix $\boldsymbol{P}^{ij}$ defined in Relation (4) as follows:

$$q_{jv} = \sum_{i=1}^{n'} \sum_{u=1}^{O_i} p_{uv}^{ij} \quad \forall j \in \{1, 2, \ldots, m+n\}, \quad \sum_{v} q_{jv} = 1 \tag{11}$$

As it was shown in Fig. 3, there are $m + n$ nodes in our considered ASC network. So, the number of rows in matrix Q denoted by j in Relation (11), is equal to $m + n$. The number of columns in matrix Q denoted by v in Relation (11) is $\mathrm{Max}_{0 \le j \le n} O_j$. It should be noted that for the rows in which $v < \mathrm{Max}_j O_j$, we have $q_{jv} = 0$ for $v < J \le \mathrm{Max}_j O_j$. In addition, the upper bound of i in Relation (11), i.e. n', is related to the number of relations with $\delta_{ij} = 1 (i, j \in \{1, 2, \ldots, m+n\})$.

In some assembly systems, flexibility is built in such a way that common tools or fixtures can be used for different variants to simplify the production process. In other words, flexible tools, common fixtures, or shared assembly procedures are assumed to treat a set of variants so that operators' selections are eliminated. In this case, the associated system's complexity reduces, because fewer selections are required. However, all the assembly processes cannot be simplified by these strategies and sometimes, flexible tools, common fixtures, or shared assembly procedures may need significant changes in product design and process planning, which are usually costly if not impossible. In order to make the relationship between product variants and process requirements to characterize the impact of flexibility, product-process association matrix Ω is defined in which variants are in the row and states are in the column (Wang et al. 2011). In general, matrix Ω for wth assembly activity at workstation l due to variety added from workstation s is defined as follows:

$$\Omega_{ls}^{w}=\begin{bmatrix}\omega_{11} & \omega_{12} & \cdots & \omega_{1,J_w}\\ \omega_{21} & \omega_{22} & \cdots & \omega_{2,J_w}\\ \vdots & \vdots & \ddots & \vdots\\ \omega_{O_j,1} & \omega_{O_j,2} & \cdots & \omega_{O_j,J_w}\end{bmatrix}_{O_j\times J_w} \tag{12}$$

where $\omega_{va}=1$ if variant v at workstation l requires wth activity to be in state a at workstation s, otherwise $\omega_{ab}=0$ and J_w represents the number of states related to wth activity. It should be noted that if selection flexibility does not exist, the number of selection alternatives is equal to the number of variants, i.e. $J_w=O_j$. But if flexibility is present and common fixtures and flexible tools can be used, the number of selection alternatives would be reduced, i.e. $J_w<O_j$. We define vector $\boldsymbol{y}_{ls}^{w}=\left[y_1,y_2,\ldots,y_{J_w}\right]$, where $\sum_a y_a=1$ and y_a denotes the probability of being wth activity in state a at workstation s due to the variants added from workstation l. Relation (13) shows how y_{ls}^{w} is calculated based on matrix Q and Relation (12).

$$y_{ls}^{w}=\mathrm{Q}_{j.}\times\Omega_{ls}^{w} \tag{13}$$

where $\mathrm{Q}_{j.}$ presents jth row of matrix Q, i.e. the mix ratio of variants at node j. Now, we are able to calculate H using vector y_{ls}^{w} obtained from Relation (13) as follows:

$$H_s^{w}=-\sum_{a=1}^{J_w} y_a\log_2 y_a \tag{14}$$

After calculating H_s^{w}s and specifying appropriate values for λ_s^{w}s, the complexity of workstation s is obtained by relation (10). Then, the total complexity of an assembly line (AS) is computed by summation of complexity of all workstations in the assembler, i.e. $AS=\sum_s SC_s$.

System complexity

When the complexity of ASC network and the complexity of assembly lines inside the assemblers are obtained according to Sects. "ASC network complexity" and "Assembly line complexity", we would be able to calculate the complexity of the whole system considering the corresponding weight factors (WF), i.e. $WF_{Network}$ and $WF_{Assembly}$, as follows:

$$\begin{aligned}&\text{Total complexity of the system}(TC)\\ &\quad=WF_{Network}\times\text{ASC network complexity }(C)\\ &\qquad+WF_{Assembly}\times\text{Complexity of assembly lines}\\ &\qquad\text{inside the assemblers }(AC)\end{aligned} \tag{15}$$

Illustrative numerical example

This section presents a numerical example to illustrate the performance of the developed model in the previous section. For this purpose, Fig. 5 shows a simple ASC network with five nodes in which nodes 1, 2, and 3 in the rectangular in the most upstream echelon are the suppliers, node 1 in the hexagonal is the intermediate sub-assembler and node 2 in the hexagonal is the final assembler. In addition, node 0 is the virtual supplier that provides the raw materials for all suppliers in the most upstream echelon. As it is shown in Fig. 5, a certain number of variants is provided in the suppliers and hence the possible combinatorial variants are assembled by the assemblers.

In this numerical example, we suppose that the customers' demand vector of the final assembler is equal to $\boldsymbol{p}_5\ =$

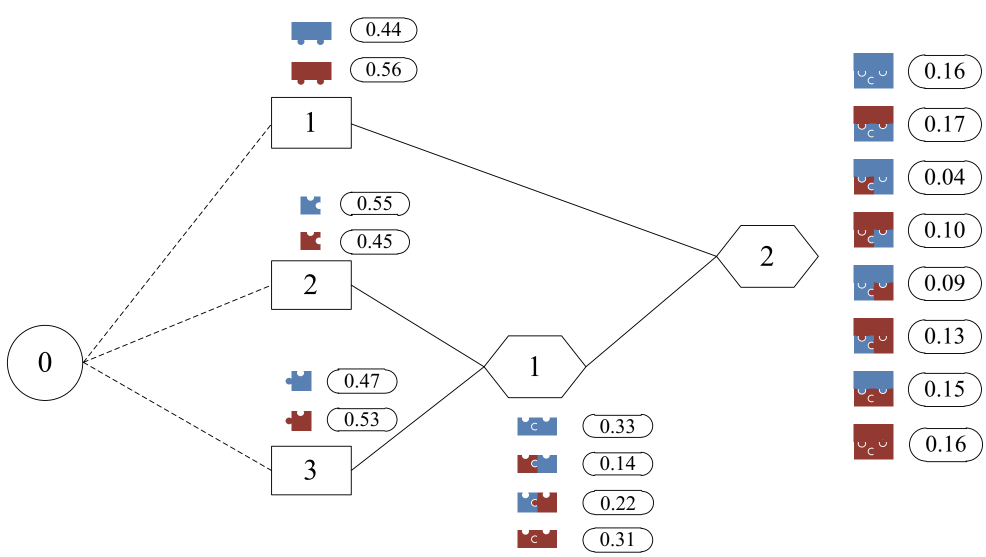


Fig. 5 The studied ASC network with 5 nodes

(0.16, 0.17, 0.04, 0.10, 0.09, 0.13, 0.15, 0.16) for eight possible product variants. If node i is a supplier of node j, it is clear that $p_{iu} = \sum_{v} p_{jv}$. Therefore, based on the vector $\boldsymbol{p}_5$, the demand vector of the other nodes can be obtained. First, matrix Δ is obtained using Relation (3) as follows:

$$\Delta = \begin{bmatrix} \delta_{00} & \delta_{01} & \delta_{02} & \delta_{03} & \delta_{04} & \delta_{05} \\ \delta_{10} & \delta_{11} & \delta_{12} & \delta_{13} & \delta_{14} & \delta_{15} \\ \delta_{20} & \delta_{21} & \delta_{22} & \delta_{23} & \delta_{24} & \delta_{25} \\ \delta_{30} & \delta_{31} & \delta_{32} & \delta_{33} & \delta_{34} & \delta_{35} \\ \delta_{40} & \delta_{41} & \delta_{42} & \delta_{43} & \delta_{44} & \delta_{45} \\ \delta_{50} & \delta_{51} & \delta_{52} & \delta_{53} & \delta_{54} & \delta_{55} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Second, for any relationship with $\delta_{ij} = 1$ in the ASC network, the corresponding matrices ($\boldsymbol{P}^{ij}$) are obtained as follows:

$$\delta_{45} = 1 \to P^{45} = \begin{bmatrix} 0.16 & 0.17 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.04 & 0.10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.09 & 0.13 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.15 & 0.16 \end{bmatrix}$$

$$\delta_{34} = 1 \to P^{34} = \begin{bmatrix} 0.33 & 0.14 & 0 & 0 \\ 0 & 0 & 0.22 & 0.31 \end{bmatrix},$$

$$\delta_{24} = 1 \to P^{24} = \begin{bmatrix} 0.33 & 0 & 0.22 & 0 \\ 0 & 0.14 & 0 & 0.31 \end{bmatrix}$$

$$\delta_{15} = 1 \to P^{15} = \begin{bmatrix} 0.16 & 0 & 0.04 & 0 & 0.09 & 0 & 0.15 & 0 \\ 0 & 0.17 & 0 & 0.10 & 0 & 0.13 & 0 & 0.16 \end{bmatrix}$$

$$\delta_{03} = 1 \to P^{03} = \begin{bmatrix} 0.47 & 0.53 \end{bmatrix},$$

$$\delta_{02} = 1 \to P^{02} = \begin{bmatrix} 0.55 & 0.45 \end{bmatrix},$$

$$\delta_{01} = 1 \to P^{01} = \begin{bmatrix} 0.44 & 0.56 \end{bmatrix}$$

According to Step 3 stated in Sect. "ASC network complexity", every matrix $\boldsymbol{P}^{ij}$ is normalized by Relation (5) where $A = 7$. For example, after normalizing matrix P^{45}, matrix $\hat{P}^{45}$ is obtained as follows. Similarly, normalized matrices $\hat{P}^{34}$, $\hat{P}^{24}$, $\hat{P}^{15}$, $\hat{P}^{03}$, $\hat{P}^{02}$, $\hat{P}^{01}$ are obtained.

Fourth, the complexity measure of any relationship of the ASC network with $\delta_{ij} = 1$ is calculated using Relation (6) as follows: $C_{45} = 0.8160$ bit, $C_{34} = 0.6767$ bit, $C_{24} = 0.6767$ bit, $C_{15} = 0.8160$ bit, $C_{03} = 0.5435$ bit, $C_{02} = 0.5429$ bit, $C_{01} = 0.5424$ bit. Finally, the complexity of the ASC network is obtained by summing the complexity measures of all relationships that is $C = C_{45} + C_{34} + C_{24} + C_{15} + C_{03} + C_{02} + C_{01} = 4.6141$ bits.

In the next step, we are going to calculate the complexity of assembly lines in two assemblers of the studied ASC network in Fig. 4 employing the method presented in Sect. "Assembly line complexity". Based on the information of Fig. 5 and Relation (11), matrix Q takes the following values:

$$Q = \begin{bmatrix} 0.44 & 0.56 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.55 & 0.45 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.47 & 0.53 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.33 & 0.14 & 0.22 & 0.31 & 0 & 0 & 0 & 0 \\ 0.16 & 0.17 & 0.04 & 0.10 & 0.09 & 0.13 & 0.15 & 0.16 \end{bmatrix}$$

Suppose four main assembly activities are recognized in the assemblers by assigning superscripts 1 to 4 at workstation s as part selection (A_{s1}), tool selection (A_{s2}), fixture selection (A_{s3}), and assembly method selection (A_{s4}). Each of these selection activities has a set of alternatives, which leads to the operator selection complexity. As it can be seen in Fig. 5, there are two sub-assemblers in the studied ASC network that each of them has its own complexity. We define AS_1 for the complexity of sub-assembler 1 and AS_2 for the complexity of sub-assembler 2. Figure 6 demonstrates the configuration of the assembly system in these assemblers. As in the assemblers two modules from upstream echelons are assembled, two workstations in a serial line are considered in each assembler. At each workstation, one corresponding module should be assembled and the operators must perform four selection activities during the assembly process (Wang et al. 2010). Thus according to Relation (9), we have the following equation for each assembler:

$$\begin{aligned} AS_i &= SC_1 + SC_2 \to AS_i \\ &= SC_{11} + SC_{22} + SC_{12} \quad (i = 1, 2) \end{aligned}$$

Then according to Relation (10), for each workstation at each assembler in this example we have:

$$\hat{P}^{45} = \begin{bmatrix} 0.0229 & 0.0243 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0057 & 0.0143 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0129 & 0.0186 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0214 & 0.0229 \end{bmatrix}$$

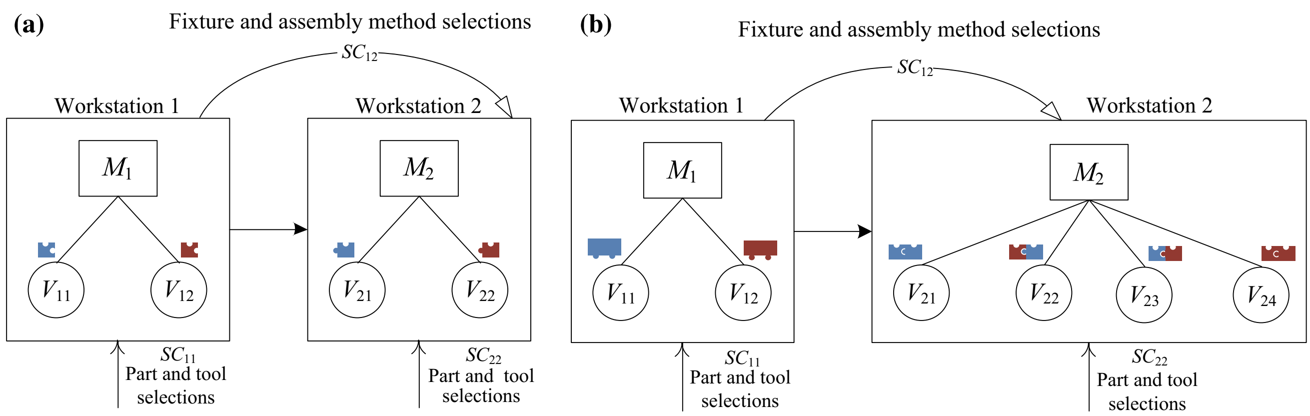


Fig. 6 **a** Configuration and complexity propagation of assembler 1; **b** Configuration and complexity propagation of assembler 2

$$SC_1 = \lambda_1^1 H_1^1 + \lambda_1^2 H_1^2,$$
$$SC_2 = \lambda_2^1 H_2^1 + \lambda_2^2 H_2^2 + \lambda_2^3 H_2^3 + \lambda_2^4 H_2^4$$

At workstation 1 of assembler 1, we consider part selection (A_{11}) and tool selection (A_{12}) for the feed complexity with the following rules:

- One of two parts, i.e. variants of module $M_1$is selected based on the customer demand.
- Use tool 2 if V_{11} is present and tool 1 if V_{12} is present.

Therefore, there are two states in the part selection and tool selection processes and the mapping relationship can be stated in a Ω matrix as follows:

$$\Omega_{11}^1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \Omega_{11}^2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

At workstation 2 of assembler 1 in addition to feed complexity, there is transfer complexity from workstation 1 with the following rules:

- One of two different fixtures is selected based on the variant of M_1 installed at workstation 1.
- One of the two different assembly methods is selected based on the variant of M_1 installed at workstation 1.

For instance using Relation (13), we calculate H_2^3 for the third activity of workstation 2 added from workstation 1 at assembler 1 as follows:

$$y_{12}^3 = \begin{bmatrix} y_1 & y_2 \end{bmatrix} = Q_{2.} \times \Omega_{12}^3$$
$$= \begin{bmatrix} 0.55 & 0.45 \end{bmatrix} \times \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0.45 & 0.55 \end{bmatrix}$$

$$\rightarrow H_2^3 = -\sum_{a=1}^{2} y_a \log_2 y_a$$
$$= -\left(0.45 \times \log_2 0.45 + 0.55 \times \log_2 0.55\right)$$
$$= 0.9928 \text{ bit}$$

It should be noted if a common assembly method is adopted at workstation 2 of assembler 1, the same method can be used for V_{21} and V_{22}. So, Ω matrix would be as follows:

$$\Omega_{12}^4 = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$$

It can simply reduce to the following form:

$$\Omega_{12}^4 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Table 1 reports more details on measuring the complexity of two assemblers in the studied example.

Finally, the total complexity of the whole studied system can be obtained by Relation (15) as follows:

$$TC = C + AC,\ AC = AS_1 + AS_2 \rightarrow TC$$
$$= 4.6141 + 0.8289 + 0.9680 = 6.411 \text{ bits}$$

Proposed algorithm to generate different ASC configurations

This section investigates how to generate different feasible configurations of ASCs while there is specific number of modules for producing a product family. The main purpose is to find the optimal ASC configuration among the generated feasible networks in terms of complexity measure for the given number of variants at the final assembler and the corresponding mix ratios of these variants. In order to reflect

Table 1 Details of the complexity calculation for two assemblers in the numerical example

Assembler	Workstation (s)	w	Selection activity	Ω matrix	y vector	H term
1	1	1	Part selection	$\Omega_{11}^1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$y_{11}^1 = \begin{bmatrix} 0.55 & 0.45 \end{bmatrix}$	$H_1^1 = 0.9928$
		2	Tool selection	$\Omega_{11}^2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	$y_{11}^2 = \begin{bmatrix} 0.45 & 0.55 \end{bmatrix}$	$H_1^2 = 0.9928$
	2	1	Part selection	$\Omega_{22}^1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$y_{22}^1 = \begin{bmatrix} 0.47 & 0.53 \end{bmatrix}$	$H_2^1 = 0.9974$
		2	Tool selection	$\Omega_{22}^2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	$y_{22}^2 = \begin{bmatrix} 0.53 & 0.47 \end{bmatrix}$	$H_2^2 = 0.9974$
		3	Fixture selection	$\Omega_{12}^3 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	$y_{12}^3 = \begin{bmatrix} 0.45 & 0.55 \end{bmatrix}$	$H_2^3 = 0.9928$
		4	Assembly method selection	$\Omega_{12}^4 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$	$y_{12}^4 = [1]$	$H_2^4 = 0$
Total complexity of assembler 1 with equal weights: $AS_1 = 0.8289$						
2	1	1	Part selection	$\Omega_{11}^1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$y_{11}^1 = \begin{bmatrix} 0.44 & 0.56 \end{bmatrix}$	$H_1^1 = 0.9896$
		2	Tool selection	$\Omega_{11}^2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	$y_{11}^2 = \begin{bmatrix} 0.56 & 0.44 \end{bmatrix}$	$H_1^2 = 0.9896$
	2	1	Part selection	$\Omega_{22}^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$y_{22}^1 = \begin{bmatrix} 0.33 & 0.14 & 0.22 & 0.31 \end{bmatrix}$	$H_2^1 = 0.9974$
		2	Tool selection	$\Omega_{22}^2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$y_{22}^2 = \begin{bmatrix} 0.55 & 0.14 & 0.31 \end{bmatrix}$	$H_2^2 = 0.9249$
		3	Fixture selection	$\Omega_{12}^3 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	$y_{12}^3 = \begin{bmatrix} 0.56 & 0.44 \end{bmatrix}$	$H_2^3 = 0.9896$
		4	Assembly method selection	$\Omega_{12}^4 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$y_{12}^4 = \begin{bmatrix} 0.44 & 0.56 \end{bmatrix}$	$H_2^4 = 0.9896$
Total complexity of assembler 2 with equal weights: $AS_2 = 0.9680$						

the real-world settings adequately, it is assumed that assembly sequence constraints exist in the assembly process. For instance in the assembly process of a laptop, the keyboard should be assembled after other required components are assembled to the main board. First, we propose a decomposition algorithm to generate all feasible ASC networks. Then based on the complexity measure obtained in the previous section, the optimal ASC network that has the minimum complexity is found.

We firstly investigate theoretically the configuration selection problem in two following special scenarios: (1) demand shares are equal for all variants at the final assembler; (2) there is one dominant variant among all the variants demanded by the customers to the final assembler. In other words, there is one dominant variant preferred by most customers and the corresponding demand share is much larger than the demand share of other variants.

Lemma 1 *Assume demand shares are equal for all the variants provided by the final assembler, i.e.,* $\boldsymbol{p}_n = (\frac{1}{On}, \frac{1}{On}, \ldots, \frac{1}{On})$. *For the sake of simplicity, suppose all the nodes in the most upstream echelon supply the same number of variants, i.e.* $O_1 = O_2 = \ldots = O_m = O$ *and hence we have* $O_n = \prod_{i=1}^{m} O_i = O^m$. *In this case, for a large number of variants produced at each node in the most upstream echelon (O) and a large number of suppliers (m), modular ASC network is more preferable than non-modular network.*

Proof At first, suppose a non-modular ASC network is considered. In this case, we have $\boldsymbol{p}_i = (\frac{1}{O}, \frac{1}{O}, \ldots, \frac{1}{O})_{1\times O}$, B_i =1 for $i = 1, 2, \ldots, m$, $\boldsymbol{p}_n = \left(\frac{1}{O^m}, \frac{1}{O^m}, \ldots, \frac{1}{O^m}\right)_{1\times O^m}$, $B_n = m$ for the final assembler (here $n = 1$) and $A = 2m$. Since we have $p_{iv} = \sum_{j=1}^{m+n} \sum_{u=1}^{O_i} p_{uv}^{ij}$, Relation (8) obtained in Sect. "ASC network complexity" can be rewritten as follows as a simplified relation:

$$C = \log_2 A - \frac{1}{A} \sum_{i=1}^{m+n} B_i \sum_{v=1}^{O_i} p_{iv} \cdot \log_2 p_{iv} \tag{16}$$

where B_i is the number of suppliers related to node $i = 1, 2, \ldots, m + n$. Now, the complexity of non-modular ASC network can be calculated as follows:

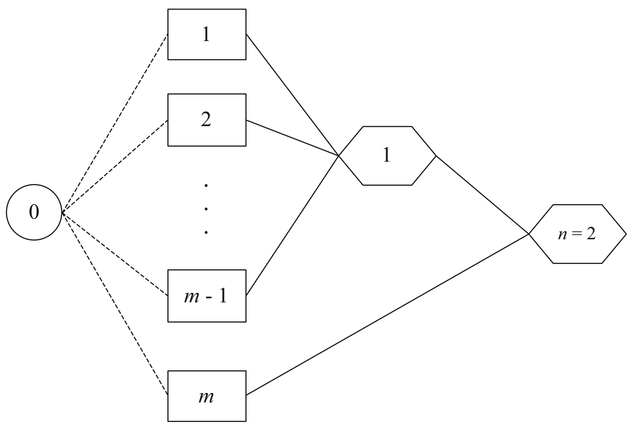


Fig. 7 The studied modular ASC network in Lemma 1 with one intermediate sub-assembler

$$C_1 = \log_2 A - \frac{1}{A}\left(\sum_{i=1}^{m}\sum_{v=1}^{O} p_{iv}\cdot\log_2 p_{iv} + m\sum_{v=1}^{O^m} p_{nv}\cdot\log_2 p_{nv}\right)$$
$$= \log_2 2m - \frac{1}{2m}\left(-m\log_2 O - m\log_2 O^m\right)$$
$$\rightarrow C_1 = \log_2 2m + \left(\frac{m+1}{2}\right)\log_2 O$$

In the second case, suppose a simple modular ASC network with only one sub-assembler for assembling the component provided by suppliers $1, 2, \ldots, m-1$ (See Fig. 7). In this case, we have $\boldsymbol{p}_i = (\frac{1}{O}, \frac{1}{O}, \ldots, \frac{1}{O})_{1\times O}$, $B_i = 1$ for $i = 1, 2, \ldots, m$, $\boldsymbol{p}_{n-1} = \left(\frac{1}{O^{m-1}}, \frac{1}{O^{m-1}}, \ldots, \frac{1}{O^{m-1}}\right)_{1\times O^{m-1}}$, $B_{n-1} = m-1$, $\boldsymbol{p}_n = \left(\frac{1}{O^m}, \frac{1}{O^m}, \ldots, \frac{1}{O^m}\right)_{1\times O^m}$, $B_n = 2$ for the final assembler (here $n = 2$) and $A = 2m+1$. So, the network complexity is obtained as follows:

$$C_2 = \log_2 A - \frac{1}{A}\left(\sum_{i=1}^{m}\sum_{v=1}^{O} p_{iv}\cdot\log_2 p_{iv} + (m-1)\sum_{v=1}^{O^{m-1}} p_{n-1,v}\cdot\log_2 p_{n-1,v} + 2\sum_{v=1}^{O^m} p_{nv}\cdot\log_2 p_{nv}\right)$$
$$\rightarrow C_2 = \log_2(2m+1) - \frac{1}{2m+1}\left(-m\log_2 O - (m-1)\log_2 O^{(m-1)} - 2\log_2 O^m\right)$$
$$= \log_2(2m+1) + \left(\frac{m^2+m+1}{2m+1}\right)\log_2 O$$

Then the difference between complexity of these two networks is obtained as follows:

$$C_2 - C_1 = \log_2(2m+1) + \left(\frac{m^2+m+1}{2m+1}\right)\log_2 O - \log_2 2m - \left(\frac{m+1}{2}\right)\log_2 O$$
$$= \log_2\left(\frac{2m+1}{2m}\right) - \left(\frac{m^2+m+1}{2m+1} - \frac{m+1}{2}\right)\log_2 O$$
$$\rightarrow C_2 - C_1 = \log_2\left(1+\frac{1}{2m}\right) - \left(\frac{m-1}{2(2m+1)}\right)\log_2 O \tag{17}$$

In Relation (17) for a given m, we have $C_2 - C_1 = \alpha - \beta\log_2 O$ where $\alpha = \log_2\left(1+\frac{1}{2m}\right) > 0$ and $\beta = \frac{m-1}{2(2m+1)} > 0$. Figures 8–9 demonstrate the behavior of this relation in terms of different m and O. As it can be seen in these figures, the obtained relation for $C_2 - C_1$ is a decreasing function of O; it would be less than zero ($C_2 - C_1 < 0$) for $m \geq 4$ and large amounts of O. In addition, the difference between C_1 and C_2 increases when the values of m and O increase. Thus for big O and m, we have $C_2 < C_1$, i.e. the complexity of the studied modular ASC network is lower than that of the non-modular ASC network. Therefore, we have shown that there is at least one modular ASC network with lower complexity than that of the non-modular ASC network when O and m are big enough. □

Lemma 2 *Suppose that among all V_n demanded by the customers to the final assembler, there is one specific dominant product, denoted as variant d, its corresponding demand share, i.e. p_{nd}, is much bigger than other variants. If the demand share of that dominant variant increases and approaches 1, i.e, $p_{nd} \rightarrow 1$, two main consequences are obtained:*
1) The complexity of the ASC network shown in Fig. 3 only depends on the supply chain structure and equals to $\log_2 A$, where A 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;
2) The optimal ASC configuration should be non-modular network.

Proof Suppose the demand vector of the final assembler is $\boldsymbol{p}_n = (p_{n1}, p_{n2}, \ldots, p_{nO_n})$. It is assumed that the dominant variant is variant d ($1 \leq d \leq O_n$) where the demand share of this variant is equal to p_{nd}. When the demand share of variant d is increased at the final assembler by the customers and approaches to 1, i.e. $p_{nd} \rightarrow 1$, the demand share of variant d at node i is also increased and approaches to 1, i.e. $p_{id} \rightarrow 1$. In addition in this case, the demand share of other variants would approach 0, i.e. $p_{iv} \rightarrow 0$ ($v = 1, 2, \ldots, O_n$ where $v \neq d$), because we know that $\sum_{v=1}^{O_n} p_{iv} = 1$. So, the ASC network complexity is obtained as follows:

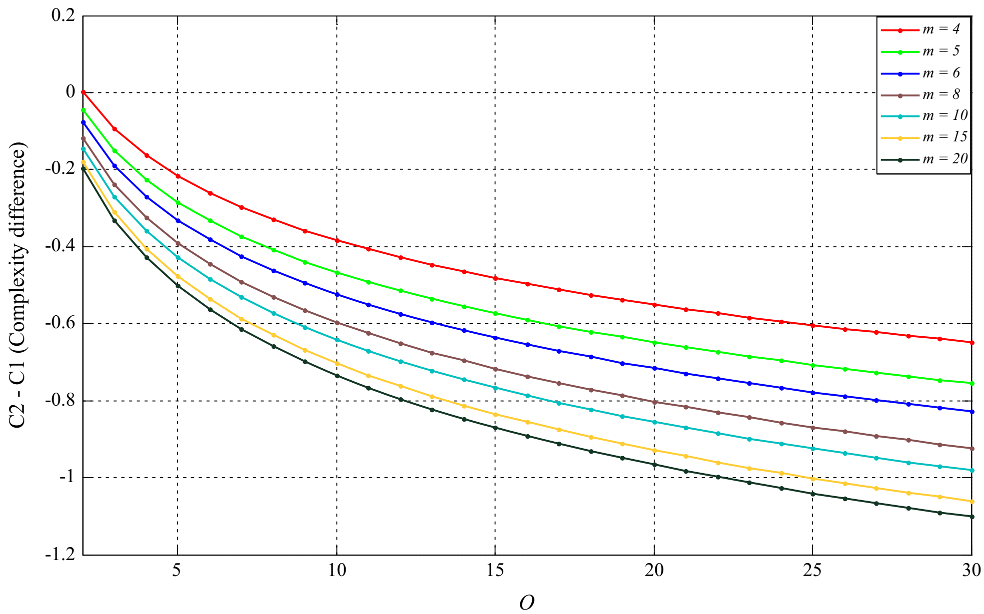


Fig. 8 The difference between complexity of two studied networks ($C_2 - C_1$) versus number of variants (O) for different number of suppliers (m)

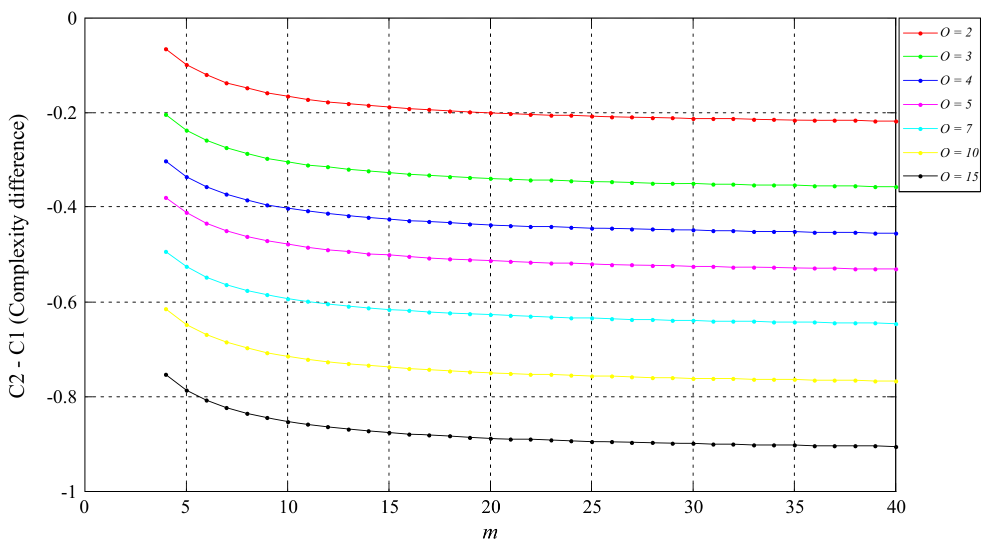


Fig. 9 The difference between complexity of two studied networks ($C_2 - C_1$) versus different number of suppliers (m) for number of variants (O)

$$C = \log_2 A - \frac{1}{A}\sum_{i=1}^{m+n} B_i \sum_{v=1}^{O_i} p_{iv} \cdot \log_2 p_{iv}$$
$$\rightarrow C = \log_2 A - \frac{1}{A}\sum_{i=1}^{m+n} B_i \left(p_{id} \cdot \log_2 p_{id} + \sum_{\substack{v=1 \\ v \neq d}}^{O_i} p_{iv} \cdot \log_2 p_{iv} \right) \tag{18}$$

Since $p_{id} \rightarrow 1$, we have $p_{id} \cdot \log_2 p_{id} = 1. \log_2 1 = 0$. However, as $p_{iv} \rightarrow 0$ ($\forall v \neq d$), we have $p_{iv} \cdot \log_2 p_{iv}$ is $0 \times \infty$ limit type. In calculus, L'Hôpital's rule can be utilized to calculate limits involving indeterminate forms using derivatives (Taylor 1952). This rule in the simplest form states that for functions f and g which are differentiable on an open interval I, if $\lim_{x\rightarrow c} f(x) = \lim_{x\rightarrow c} g(x) = 0$ or $\pm\infty$, $\lim_{x\rightarrow c} \frac{f'(x)}{g'(x)}$ exists, and $g'(x) \neq 0$ for $x \in I - \{c\}$, then the following relation can be written:

$$\lim_{x\rightarrow c} \frac{f(x)}{g(x)} = \lim_{x\rightarrow c} \frac{f'(x)}{g'(x)} \tag{19}$$

Using Relation (19), we have:

$$\begin{aligned}\lim_{p_{iv}\to 0} p_{iv}\cdot\log_2 p_{iv} &= \lim_{p_{iv}\to 0}\frac{\log_2 p_{iv}}{1/p_{iv}}\\ &= \lim_{p_{iv}\to 0}\frac{(\log_2 p_{iv})'}{(1/p_{iv})'}\\ &= \lim_{p_{iv}\to 0}\frac{1/(\ln 2\times p_{iv})}{-1/(p_{iv})^2}\\ &= \lim_{p_{iv}\to 0} -\frac{p_{iv}}{\ln 2} = 0.\end{aligned}$$

By substituting the obtained values in Relation (18), the complexity of ASC network is equal to $C = \log_2 A$, where A is the number of total arcs in the network. Therefore in this case, the ASC network complexity is only dependent on the number of arcs. As it is clear and can be seen in Fig. 3, A is obtained by the following relationship without considering the arcs to the customers:

$$A = 2m + n - 1 \tag{20}$$

where m and n denote the number of suppliers in the most upstream echelon and the number of assemblers, respectively. Since m is fixed, in order to obtain the minimum amount of A in Relation $C = \log_2 A$, we should select the minimum n. As it is shown in Fig. 1, non-modular ASC network has the minimum number of assemblers that is $n = 1$ and the corresponding ASC network complexity is $C = \log_2 2m$, because in this case $A = 2m$. Thus, it was demonstrated that in the scenario of one dominant variant, the optimal ASC configuration is non-modular. □

If it is recognized that modular configuration is more beneficial than non-modular, a method is required to generate different networks in order to connect nodes in the upstream nodes to the final assembler. Furthermore, there are different possible alternatives for each configuration due to different locations of suppliers in the most upstream echelon. For example, suppose there are four nodes in the most upstream echelon as the initial suppliers in which each supplier provides one of the modules M_1, M_2, M_3, and M_4. In this case, five different configurations can be generated as shown in Fig. 10 where the corresponding alternatives for configurations No. 2 and No. 4 have been demonstrated. As it can be seen in this figure, configurations No. 2 and No. 4 have four and three possible ASC alternatives, respectively.

In order to obtain all possible ASC configurations, at first we need to define symbols "{" and "}" for assembling modules together. For instance, sub-assembly $\{M_1M_3M_4\}$ indicates that modules M_1, M_3 and M_4 are assembled together at one node but sub-assembly $\{\{M_1M_3\}M_4\}$ indicates that at first, modules M_1 and M_3 are assembled at node M_1M_3 and then module M_4 is added to them at another node.

Webbink and Hu (2005) employ a similar definition to represent the system configurations in the manufacturing system design area and develop a decomposition algorithm to generate the manufacturing system configurations consisting n workstations based on their definition. In this paper, we utilize the similar definition to represent ASC networks in which one pair of braces denotes an assembly relationship. Starting from the most upstream echelon of the supply chain and moving to the final assembler, a pair of braces is added when an assembly relationship is required. This process is repeated until the final assembler is reached. For example, in the first alternative of configuration No. 4 in Fig. 10a, starting from the most upstream echelon with four suppliers and moving forward, modules M_1 and M_2 are assembled together at node M_1M_2 and one pair of braces is added, so sub-assembly $\{M_1M_2\}$ is obtained. In a similar way, sub-assembly $\{M_3M_4\}$ is obtained. Moving forward, sub-assemblies $\{M_1M_2\}$ and $\{M_3M_4\}$ are assembled together at the final assembler and one more pair of braces is added. Therefore, $\{\{M_1M_2\}\{M_3M_4\}\}$ is obtained at the final assembler of the above-mentioned configuration.

After defining this method for ASC, an iterative decomposition algorithm is proposed to generate all possible ASC candidates when the number of nodes in the most upstream echelon is given. As mentioned before, in order to reflect the real-world situation, it is assumed that there is assembly sequence constraint (precedence constraint) to obtain the final product. Therefore considering this assumption, the proposed algorithm has the following steps:

Step 1- Based on the number of modules at the final product, generate all sets of sub-assemblies and modules, which can be produced by the nodes in all possible ASC networks. In this case, if there are m suppliers, we can generate $\binom{m}{1} + \binom{m}{2} + \cdots + \binom{m}{m-1}$ sets of sub-assemblies and modules. For example, if there are four modules M_1, M_2, M_3, and M_4 similar to Fig. 10, the generated sub-assemblies and modules are $\{M_1\}$, $\{M_2\}$, $\{M_3\}$, $\{M_4\}$, $\{M_1M_2\}$, $\{M_1M_3\}$, $\{M_1M_4\}$, $\{M_2M_3\}$, $\{M_2M_4\}$, $\{M_3M_4\}$, $\{M_1M_2M_3\}$, $\{M_1M_2M_4\}$, $\{M_1M_3M_4\}$.

Step 2- List all possible assembly combinations of sub-assemblies and modules generated from Step 1, through which the final product can be obtained after the first decomposition and then one pair of braces is added. In the simple example presented in Step 1, 14 possible combinations are $\{\{M_1M_2M_3\}\{M_4\}\}$, $\{\{M_1M_2M_4\}\{M_3\}\}$, $\{\{M_1M_3M_4\}\{M_2\}\}$, $\{\{M_2M_3M_4\}\{M_1\}\}$, $\{\{M_1M_2\}\{M_3\}\{M_4\}\}$, $\{\{M_1M_3\}\{M_2\}\{M_4\}\}$, $\{\{M_1M_4\}\{M_2\}\{M_3\}\}$, $\{\{M_2M_3\}\{M_1\}\{M_4\}\}$, $\{\{M_2M_4\}\{M_1\}\{M_3\}\}$, $\{\{M_3M_4\}\{M_1\}\{M_2\}\}$, $\{\{M_1M_2\}\{M_3M_4\}\}$, $\{\{M_1M_3\}\{M_2M_4\}\}$, $\{\{M_1M_4\}\{M_2M_3\}\}$, $\{\{M_1\}\{M_2\}\{M_3\}\{M_4\}\}$. As it is clear, in the generated combinations in this step, only one stage of decomposition has been performed.

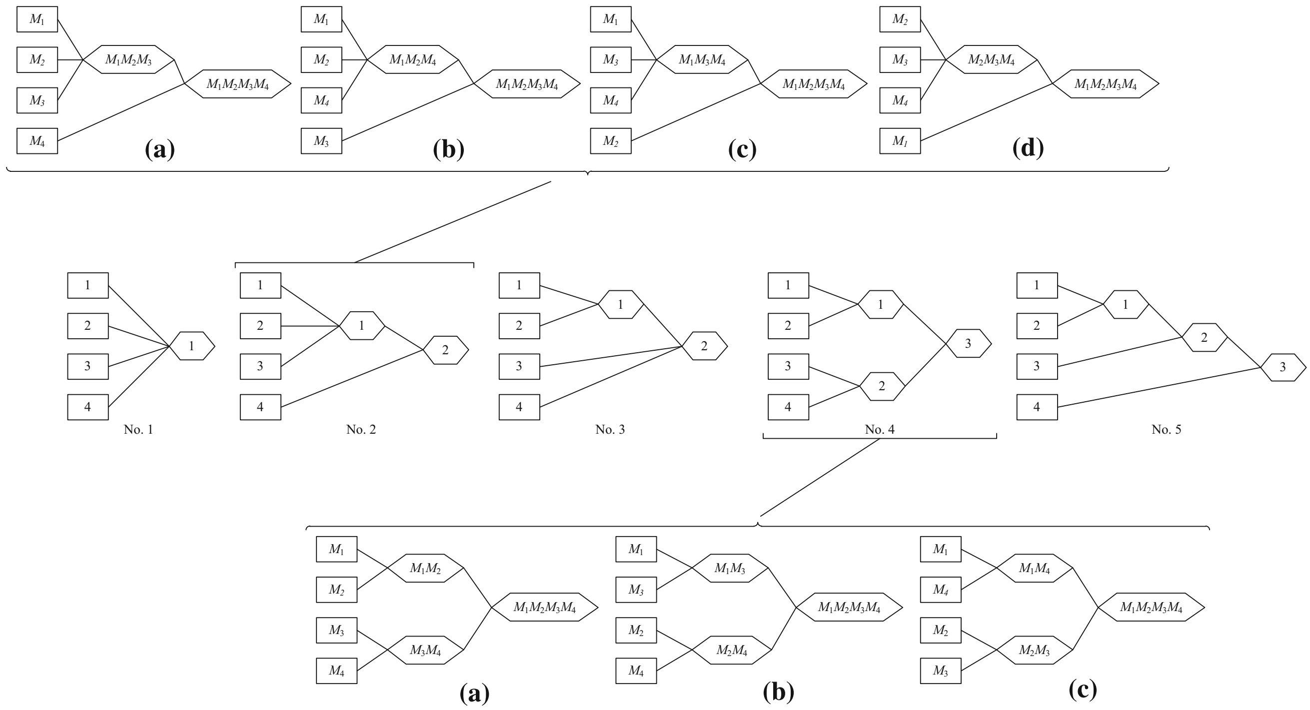


Fig. 10 The possible ASC network with four suppliers and the corresponding alternatives for configurations No. 2 and No. 4

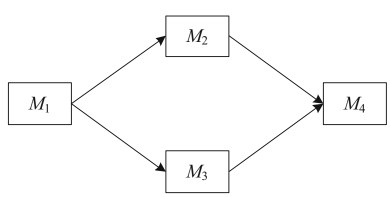


Fig. 11 The precedence diagram for the illustrative example with 4 modules

Step 3- Based on the available precedence constraints, check the feasibility of the generated combinations in Step 2 and delete the infeasible ones that do not meet the constraints.
Step 4- For feasible combinations obtained from Step 3, check the cardinality of each inner braces. If there is the inner braces with the cardinality more than one, a sub-assembly relationship is required for that inner braces. In this case, that sub-assembly is treated as the final product in Step 1. Thus, go back to Step 1 and repeat the process until the cardinality of all inner braces equals one, which means no more decomposition in sub-assemblies is required.

In order to illustrate the proposed algorithm, the simple above-mentioned example with four modules (suppliers) is employed. For this purpose, suppose a precedence diagram as shown in Fig. 11 in which modules M_2 and M_3 cannot be assembled until module M_1 is assembled. In addition, module M_4 must be assembled after modules M_2 and M_3 are assembled. Table 2 reports the details of above-mentioned algorithm. As it can be seen in Table 2, the possible assembly combinations have been presented in the first column (Stage 1), as mentioned in Step 1. Then, the feasibility check of the combinations is performed and the corresponding result for each combination is reported in "Status" column as "Feasible" or "Infeasible". In this way, the infeasible combinations are not decomposed, so the efficiency of the proposed algorithm increases. The second decomposition in Stage 2 is performed with four feasible combinations in which for the second and third ones, there are two decompositions. In each stage, the cardinality of the inner braces is checked and the decomposition continues until all cardinalities equal one. In the studied example in this section, the decomposition process continues until Stage 3 in some combinations. Last column of Table 2 shows the corresponding configurations with generated combinations.

Selection of optimal ASC network

In this section, we are going to find the optimal assembly supply chain under general demands, while the number of variants at the final assembler and the mix ratios of these variants are given. For this purpose, after obtaining the feasible ASC candidates using the decomposition iterative algorithm presented in Sect. "Proposed algorithm to generate different ASC configurations", the optimal network can be obtained comparing the total complexity values of these candidates. For better understanding,

Table 2 The iterative decomposition method to generate feasible ASC networks

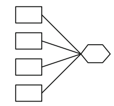
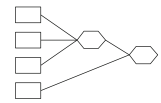
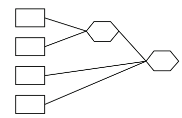
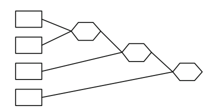
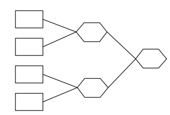
Decomposition of final product $M_1M_2M_3M_4$			Status	Configuration
Stage 1	Stage 2	Stage 3		
$\{\{M_1\}\{M_2\}\{M_3\}\{M_4\}\}$	-		Feasible	
$\{\{M_1M_2M_3\}\{M_4\}\}$	$\{\{M_1\}\{M_2\}\{M_3\}\{M_4\}\}$		Feasible	
$\{\{M_1M_2M_4\}\{M_3\}\}$	-		Infeasible	
$\{\{M_1M_3M_4\}\{M_2\}\}$	-		Infeasible	
$\{\{M_2M_3M_4\}\{M_1\}\}$	-		Infeasible	
$\{\{M_1M_2\}\{M_3\}\{M_4\}\}$	$\{\{\{M_1\}\{M_2\}\}\{M_3\}\{M_4\}\}$		Feasible	
	$\{\{\{M_1M_2\}\{M_3\}\}\{M_4\}\}$	$\{\{\{\{M_1\}\{M_2\}\}\{M_3\}\}\{M_4\}\}$		
$\{\{M_1M_3\}\{M_2\}\{M_4\}\}$	$\{\{\{M_1\}\{M_3\}\}\{M_2\}\{M_4\}\}$		Feasible	
	$\{\{\{M_1M_3\}\{M_2\}\}\{M_4\}\}$	$\{\{\{\{M_1\}\{M_3\}\}\{M_2\}\}\{M_4\}\}$		
$\{\{M_1M_4\}\{M_2\}\{M_3\}\}$	-		Infeasible	
$\{\{M_2M_3\}\{M_1\}\{M_4\}\}$	-		Infeasible	
$\{\{M_2M_4\}\{M_1\}\{M_3\}\}$	-		Infeasible	
$\{\{M_3M_4\}\{M_1\}\{M_2\}\}$	-		Infeasible	
$\{\{M_1M_2\}\{M_3M_4\}\}$	-		Infeasible	
$\{\{M_1M_3\}\{M_2M_4\}\}$	-		Infeasible	
$\{\{M_1M_4\}\{M_2M_3\}\}$	-		Infeasible	

Table 3 Comparison of the total complexity among feasible configurations

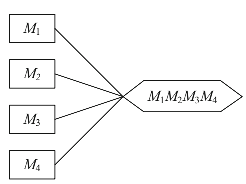
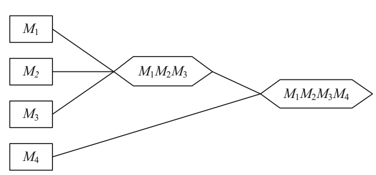
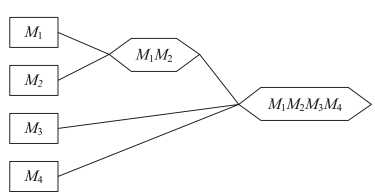
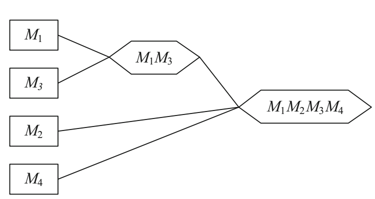
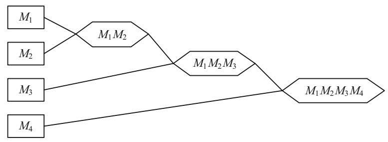
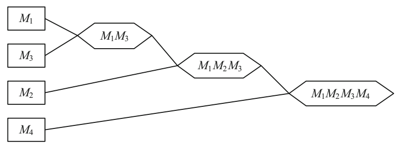
No.	Feasible configuration	Network complexity (C)	Assembly line complexity (AC)	Total complexity (TC)
1		5.2929	1.9954	7.2883
2		5.3532	1.8302	7.1834
3		5.2384	1.8095	7.0479
4		5.2370	1.8127	7.0497

Table 3 continued

No.	Feasible configuration	Network complexity (C)	Assembly line complexity (AC)	Total complexity (TC)
5		5.3903	2.3635	7.7538
6		5.4015	2.3851	7.7866

the simple example with four modules presented in Sect. “Proposed algorithm to generate different ASC configurations” is used here. In this example according to Table 2, there are 4 feasible ASC networks. Suppose there are 16 variants at the final assembler with the demand vector as $\boldsymbol{p}_i = (0.07, 0.00, 0.02, 0.12, 0.10, 0.14, 0.09, 0.05, 0.03, 0.09, 0.01, 0.04, 0.06, 0.00, 0.08, 0.10)$ where each supplier provides one module and each module has two variants. Table 3 reports the details of measuring the total complexity for the feasible ASC networks based on descriptions of Sects. “Modeling and measuring of the complexity” and “Proposed algorithm to generate different ASC configurations” in which TC is obtained with equal weights. As it can be seen in Table 3, configuration No. 3 has the minimum total complexity among 6 feasible configurations. It should be noted that the total complexity in the last column of Table 3 is calculated using Relation (15) with equal weight factors.

Conclusions and future studies

This paper dealt with the complexity modeling of assembly supply chains (ASCs) based on Shannon’s information entropy. For this purpose, after calculating the complexity of ASC network and the complexity of assembly lines inside the assemblers, the total complexity of the whole system was measured. The definition of complexity measure in this paper was developed by integrating the detailed information of the supply chain configuration, the number of variants at each node of ASC network and the corresponding mix ratios of these variants. In addition, an algorithm was proposed to generate different feasible configurations of ASCs while there are assembly sequence constraints in the assembly process and specific number of variants for a product family. Finally, the proposed complexity measure was used to find the optimal ASC configuration among the generated feasible networks for the given number of variants and mix ratios at the final assembler.

This paper studied forward supply chain networks. Future research can be focused on complexity modeling of reverse supply chain networks and disassembly operations. Measuring the costs associated with increase in the structural complexity with respect to different cost structures for the elements that contribute to the complexity can be another challenge for future studies.

References

Baud-Lavigne, B., Bassetto, S., & Agard, B. (2014). A method for a robust optimization of joint product and supply chain design. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-014-0908-5.

Blecker, T., Kersten, W., Meyer, C.M. (2005). Development of an approach for analyzing supply chain complexity. In *Mass Customization: Concepts–Tools–Realization. Proceedings of the International Mass Customization Meeting* (pp. 47–59).

Borda, M. (2011). *Fundamentals in information theory and coding* (Vol. 6). New York: Springer.

Chiu, M.-C., & Okudan, G. (2014). An investigation on the impact of product modularity level on supply chain performance metrics: An industrial case study. *Journal of Intelligent Manufacturing*, *25*(1), 129–145. doi:10.1007/s10845-012-0680-3.

Deshmukh, A. V., Talavage, J. J., & Barash, M. M. (1998). Complexity in manufacturing systems, Part 1: Analysis of static complexity. *IIE Transactions*, *30*(7), 645–655.

ElMaraghy, W., ElMaraghy, H., Tomiyama, T., & Monostori, L. (2012). Complexity in engineering design and manufacturing. *CIRP Annals-Manufacturing Technology*, *61*(2), 793–814.

Frizelle, G., & Woodcock, E. (1995). Measuring complexity as an aid to developing operational strategy. *International Journal of Operations & Production Management*, *15*(5), 26–39.

Fujita, K., Amaya, H., & Akai, R. (2013). Mathematical model for simultaneous design of module commonalization and supply chain configuration toward global product family. *Journal of Intelligent Manufacturing*, *24*(5), 991–1004. doi:10.1007/s10845-012-0641-x.

Hamta, N., Akbarpour Shirazi, M., & Fatemi Ghomi, S. M. T. (2015). A bi-level programming model for supply chain network optimization with assembly line balancing and push–pull strategy. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*. doi:10.1177/0954405414564406.

Hu, S. J., Zhu, X., Wang, H., & Koren, Y. (2008). Product variety and manufacturing complexity in assembly systems and supply chains. *CIRP Annals-Manufacturing Technology*, *57*(1), 45–48.

İnkaya, T., & Akansel, M. (2015). Coordinated scheduling of the transfer lots in an assembly-type supply chain: A genetic algorithm approach. *Journal of Intelligent Manufacturing*. doi:10.1007/s10845-015-1041-9.

Milgate, M. (2001). Supply chain complexity and delivery performance: An international exploratory study. *Supply Chain Management: An International Journal*, *6*(3), 106–118.

Modrak, V., & Marton, D. (2012). Modelling and complexity assessment of assembly supply chain systems. *Procedia Engineering*, *48*, 428–435.

Modrak, V., & Marton, D. (2013). Structural complexity of assembly supply chains: A theoretical framework. *Procedia CIRP*, *7*, 43–48.

Papakostas, N., Efthymiou, K., Mourtzis, D., & Chryssolouris, G. (2009). Modelling the complexity of manufacturing systems using nonlinear dynamics approaches. *CIRP Annals-Manufacturing Technology*, *58*(1), 437–440.

Schuh, G., Monostori, L., Csáji, B. C., & Döring, S. (2008). Complexity-based modeling of reconfigurable collaborations in production industry. *CIRP Annals-Manufacturing Technology*, *57*(1), 445–450.

Shannon, C. E. (2001). A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, *5*(1), 3–55.

Sivadasan, S., Efstathiou, J., Calinescu, A., & Huatuco, L. H. (2006). Advances on measuring the operational complexity of supplier-customer systems. *European Journal of Operational Research*, *171*(1), 208–226.

Suh, N. P. (1999). A theory of complexity, periodicity and the design axioms. *Research in Engineering Design*, *11*(2), 116–132.

Suh, N. P. (2005). Complexity in engineering. *CIRP Annals-Manufacturing Technology*, *54*(2), 46–63.

Su, J. P., Lin, Y., & Lee, V. (2012). Component commonality in closed-loop manufacturing systems. *Journal of Intelligent Manufacturing*, *23*(6), 2383–2396. doi:10.1007/s10845-010-0485-1.

Taylor, A. E. (1952). L'Hospital's rule. *The American Mathematical Monthly, 59*(1), 20–24.

Wang, H., Ko, J., Zhu, X., & Hu, S. J. (2010). A complexity model for assembly supply chains and its application to configuration design. *Journal of Manufacturing Science and Engineering*, *132*(2), 21005.

Wang, H., Zhu, X., Wang, H., Hu, S. J., Lin, Z., & Chen, G. (2011). Multi-objective optimization of product variety and manufacturing complexity in mixed-model assembly systems. *Journal of Manufacturing Systems*, *30*(1), 16–27.

Webbink, R. F., & Hu, S. J. (2005). Automated generation of assembly system-design solutions. *Automation Science and Engineering, IEEE Transactions on*, *2*(1), 32–39.

Wiendahl, H.-P., & Scholtissek, P. (1994). Management and control of complexity in manufacturing. *CIRP Annals-Manufacturing Technology*, *43*(2), 533–540.

Wilding, R. D. (1998). Chaos theory: Implications for supply chain management. *International Journal of Logistics Management, The*, *9*(1), 43–56.

Zeltzer, L., Limère, V., Van Landeghem, H., Aghezzaf, E.-H., & Stahre, J. (2013). Measuring complexity in mixed-model assembly workstations. *International Journal of Production Research*, *51*(15), 4630–4643.

Zhu, X., Hu, S. J., Koren, Y., & Marin, S. P. (2008). Modeling of manufacturing complexity in mixed-model assembly lines. *Journal of Manufacturing Science and Engineering*, *130*(5), 51013.