PRODUCT VARIETY INDUCED COMPLEXITY AND ITS IMPACT ON MIXED-MODEL ASSEMBLY SYSTEMS AND SUPPLY CHAINS

by

Hui Wang

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy (Mechanical Engineering and Industrial and Operations Engineering) in The University of Michigan 2010

Doctoral Committee:
Professor Shixin Jack Hu, Co-Chair
Associate Professor Göker Aydın, Co-Chair
Professor Yoram Koren
Associate Professor Mark Van Oyen
Jeffrey Alden, General Motors
Dedicated to my family who have loved and supported me
ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my committee members, my colleagues, my friends, my family, and my Lord for their guidance, help, love and support. Without them, I would never have been able to finish this dissertation.

First and foremost, I would like to thank my co-advisors, Professor S. Jack Hu and Professor Göker Aydın, for the extraordinary vision, knowledge and guidance they provided me throughout my time at the University of Michigan. I feel overwhelmingly blessed and forever grateful, being their student. Professor Hu has given me tremendous help during my academic journey from Tianjin to Ann Arbor. He inspired me to explore this complexity research in the supply chain area and has provided the invaluable insights and continuous encouragement whenever challenges arose. Word by word, line by line, he taught me how to think logically and express concisely and precisely. Professor Aydın both taught and demonstrated to me how to be a qualified and independent researcher. I will never forget the countless hours he spent reviewing the technical details of my research, checking every step of the proofs, editing every sentence and punctuation mark in my papers, etc. Even though I made many mistakes as I learned and grew, his patience with me during this time strengthened me to persevere when I had nearly given up.

I would also like to extend my thanks to my other committee members, Professor Yoram Koren, Professor Mark Van Oyen and Dr. Jeffrey Alden. Professor Koren gave me the wonderful opportunity to work in the Engineering Research Center (ERC) over the past five years, which has impacted all my work. As an academic advisor as well as a life mentor, Professor Koren has taught me how to be an outstanding engineer and a good citizen. His inspiration has simply been invaluable to me. Professor Van Oyen is one of the best experts in Operations Research. I appreciate his valuable feedback and constructive suggestions to
this research. Dr. Alden has been working in the production department at General Motors R&D for many years. His industrial experience and expertise proved an integral part to this research.

I gratefully acknowledge the financial support from the Engineering Research Center for Reconfigurable Manufacturing Systems (ERC/RMS) of the National Science Foundation under Award Number EEC-9529125 and the National Science Foundation under Award Number CMMI-0825438, both at The University of Michigan.

I also want to thank my friends and colleagues in Engineering Research Center (ERC) and General Motor Collaborative Research Lab (GMCRL), with whom I had the privilege to work and spend time together, including but not limited to Jeonghan Ko, Guosong Lin, Luis Izquierdo, Hao Du, Jianpeng Yue, Xiaowei Zhu, April Brian, Yuanyuan Zhou, Liang Zhou, Jingjing Li, Sha Li, Hai Nguyen, Shawn Lee, Chenhui Shao, Jialu Liu, He Wang, Tae Kim, Hui Wang, etc. I am grateful for their friendship which made my life easier, happier, and warmer during the long-lasting Michigan winters.

Here I also want give my sincere thanks to Larry Champoux, who has been my friend, brother and spiritual mentor in Christ for more than five years. Thank you, Larry, for every Bible study we had together and every prayer you gave to me. I feel so blessed to have you in my life encouraging me and helping me grow in Christ. I also want to thank my beloved brothers and sisters in Grace Student Fellowship and Ann Arbor Chinese Christian Church (AACCC), including but not limited to Brent Hoover, Harold Yan, Mike Yu, Jessica Duan, Robbie Hopkins, Qinyi Yan, Qin Li, Zhihua Zhang, Minghua Wang, Yin Song, Xiaotong Zhang, Shenquan Wang, Benbo Song, Yibin Jiang, Weiwei Li, Yanhui Wang, Yunyan Wang, Yu Qiao, Yilun Wu, Wei Gu, Suyi Li, Wei Li, Zeyu Li, Shifang Li, Jinjin Ma, Baowen Du.

I would like to express my thanks to David Green. Thank you for the never-ending support, continuous encouragement and unconditional love. I feel so grateful and honored to be a part of Green’s family. You make me want to be a better man. I also want to give the thanks to my biggest fan, James Conway. I feel forever grateful to you for your understanding, tolerance and patience. Thank you for giving me strength when I was weak, standing by me when I was down, giving me faith when I wanted to give up, lifting me up when I could not reach. My world is a better place because of you.
In addition, my greatest gratitude goes to my family for their unconditional love and support. None of this would have been possible without the love and support from them. My parents, Qirui Wang and Jizhen Cai, are always proud of me and believe in me. From them, I received the most encouragement and strength in the difficult times. My sister, Fang Wang, and my brother-in-law, Jian Lin, have also provided their best to support me. My 8-month niece, Xinyi Lin, has brought a lot of unforgettable joy and laugh to my life.

Last but certainly not least, I want to give my greatest thanks to my savior and my Lord, Jesus Christ. Thank you for letting me know you, the most precious thing in the world. Thank you for sacrificing yourself for me so that I can be forgiven and have hope in my life. Thank you for the never-ceasing and unconditional love. I will never forget the promise from you. “For I know the plans I have for you,” declares the LORD, “plans to prosper you and not to harm you, plans to give you hope and a future. Then you will call upon me and come and pray to me, and I will listen to you. You will seek me and find me when you seek me with all your heart.” Jeremiah 29:11-13

To everyone I will always be in debt. Thank you all!
# TABLE OF CONTENTS

DEDICATION .................................................................................................................. ii

ACKNOWLEDGEMENTS ................................................................................................. iii

LIST OF FIGURES ........................................................................................................ viii

LIST OF TABLES ........................................................................................................... x

ABSTRACT ..................................................................................................................... xi

CHAPTER

1 INTRODUCTION ........................................................................................................ 1
  1.1 Motivation ........................................................................................................... 1
    1.1.1 Mixed-model Assembly Systems ................................................................. 2
    1.1.2 Modular Assembly Supply Chains ................................................................. 3
  1.2 Research Objective and Tasks ........................................................................... 4
  1.3 Organization of the Dissertation ........................................................................ 5

2 PRODUCT VARIETY INDUCED COMPLEXITY IN MIXED-MODEL ASSEMBLY SYSTEMS AND ITS IMPACT ON THROUGHPUT ........................................ 8
  2.1 Introduction ....................................................................................................... 9
  2.2 Model Description ............................................................................................ 12
  2.3 Complexity of MASs with Different Configurations ...................................... 17
    2.3.1 Complexity of a MAS with Serial Configuration ....................................... 17
    2.3.2 Complexity of a MAS with Parallel Configuration ................................. 18
    2.3.3 Complexity of a MAS with Hybrid Configuration ................................. 19
  2.4 Throughput Model of MASs with Different Configurations .......................... 20
    2.4.1 Reliability Model for a Mixed-model Assembly Station .......................... 20
    2.4.2 Throughput Model for MASs .................................................................... 23
  2.5 Comparison of MASs with Different Configurations ...................................... 23
  2.6 Product Variant Differentiation in MASs ......................................................... 31
  2.7 Conclusions ....................................................................................................... 37

3 A COMPLEXITY MEASURE FOR ASSEMBLY SUPPLY CHAINS AND ITS RELATIONSHIP TO COST ........................................................................ 41
  3.1 Introduction ....................................................................................................... 42
  3.2 Literature Review ............................................................................................. 44
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>An example of mixed-model assembly system (MAS)</td>
<td>10</td>
</tr>
<tr>
<td>2.2</td>
<td>Different configurations for MASs with four assembly stations</td>
<td>11</td>
</tr>
<tr>
<td>2.3</td>
<td>Combinational assembly of four modules and the demand vector relationships between $Q^{(3,4)}$ and $Q$</td>
<td>13</td>
</tr>
<tr>
<td>2.4</td>
<td>Two different MASs share the same configuration. MASs in (a) and (b) have different demand fraction of variants at parallel stations</td>
<td>15</td>
</tr>
<tr>
<td>2.5</td>
<td>An example to illustrate the aggregation algorithm of calculating the complexity of a MAS with hybrid configuration</td>
<td>19</td>
</tr>
<tr>
<td>2.6</td>
<td>Distribution algorithm to generate system configurations of four stations</td>
<td>24</td>
</tr>
<tr>
<td>2.7</td>
<td>Complexity of the six configurations under evenly distributed demands</td>
<td>25</td>
</tr>
<tr>
<td>2.8</td>
<td>Throughput of the six configurations as a function of $\delta$ under evenly distributed demands</td>
<td>27</td>
</tr>
<tr>
<td>2.9</td>
<td>Complexity of the six configurations under generally distributed demands</td>
<td>28</td>
</tr>
<tr>
<td>2.10</td>
<td>Throughput of the six configurations as a function of $\delta$ under generally distributed demands</td>
<td>29</td>
</tr>
<tr>
<td>2.11</td>
<td>There are six MASs that have the same configuration as configuration (b)</td>
<td>30</td>
</tr>
<tr>
<td>2.12</td>
<td>Throughput of MASs within configuration (b) as the function of $\delta$</td>
<td>30</td>
</tr>
<tr>
<td>2.13</td>
<td>An example to illustrate using differentiating the production of variants to reduce the complexity of MASs and increase the throughput</td>
<td>31</td>
</tr>
<tr>
<td>3.1</td>
<td>Relationship between the demand vectors of upstream and downstream nodes</td>
<td>48</td>
</tr>
<tr>
<td>3.2</td>
<td>Two assembly supply chains with the same number of nodes, but distinct configurations</td>
<td>50</td>
</tr>
<tr>
<td>3.3</td>
<td>An example where complexity and cost are inconsistent</td>
<td>58</td>
</tr>
<tr>
<td>3.4</td>
<td>Two new levels of Factor ‘B,’ the number of echelons</td>
<td>63</td>
</tr>
<tr>
<td>3.5</td>
<td>Histogram of cost difference</td>
<td>64</td>
</tr>
<tr>
<td>3.6</td>
<td>Histogram of cost differences in the modified numerical study</td>
<td>65</td>
</tr>
<tr>
<td>3.7</td>
<td>Prototypical configurations for non-modular vs. modular assembly supply chains</td>
<td>66</td>
</tr>
<tr>
<td>3.8</td>
<td>Complexity difference between the non-modular supply chain and the modular in four different scenarios</td>
<td>83</td>
</tr>
<tr>
<td>3.9</td>
<td>Cost difference between the non-modular supply chain and the modular in four different scenarios</td>
<td>85</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Description of serial, parallel and hybrid configurations</td>
<td>14</td>
</tr>
<tr>
<td>2.2</td>
<td>An example to illustrate using differentiating the production of variants to reduce the complexity of MASs and increase the throughput</td>
<td>32</td>
</tr>
<tr>
<td>2.3</td>
<td>Numerical study results of using differentiating the production of variants to reduce the complexity of MASs and increase the throughput</td>
<td>36</td>
</tr>
<tr>
<td>3.1</td>
<td>A $2^5$ full factorial design of DOE</td>
<td>60</td>
</tr>
<tr>
<td>3.2</td>
<td>Estimated effects, t-Statistics and p-Value (%) of the experiment shown in Table 3.1</td>
<td>61</td>
</tr>
<tr>
<td>3.3</td>
<td>Cost difference statistics (%)</td>
<td>63</td>
</tr>
<tr>
<td>3.4</td>
<td>Cost difference statistics (%) in the modified numerical study</td>
<td>64</td>
</tr>
<tr>
<td>3.5</td>
<td>The percentage of inconsistencies for each of the 32 value combinations of the five factors shown in Table 3.1</td>
<td>92</td>
</tr>
<tr>
<td>3.6</td>
<td>The percentage of inconsistencies for each of the 32 value combinations of the five factors shown in Table 3.1, except that the number of suppliers is now four or five (as opposed two or three)</td>
<td>93</td>
</tr>
<tr>
<td>3.7</td>
<td>Estimated effects, t-Statistics and p-Value (%) of the experiment shown in Table 3.6</td>
<td>93</td>
</tr>
<tr>
<td>3.8</td>
<td>The percentage of inconsistencies for each of the 32 value combinations of the five factors shown in Table 3.1, except that the cost of the downstream echelon is now four or five times as large as the sum of cost coefficients of the nodes in the upstream echelon (as opposed to being the same as the sum or twice as large)</td>
<td>94</td>
</tr>
<tr>
<td>3.9</td>
<td>Estimated effects, t-Statistics and p-Value (%) of the experiment shown in Table 3.8</td>
<td>94</td>
</tr>
<tr>
<td>3.10</td>
<td>The percentage of inconsistencies for each of the 32 value combinations of the five factors shown in Table 3.1, except that the number of suppliers is fixed at two and the factor of the number of suppliers is replaced with number of echelons (number of echelons could either be two or three)</td>
<td>95</td>
</tr>
<tr>
<td>3.11</td>
<td>Estimated effects, t-Statistics and p-Value (%) of the experiment shown in Table 3.10</td>
<td>95</td>
</tr>
</tbody>
</table>
ABSTRACT

PRODUCT VARIETY INDUCED COMPLEXITY AND ITS IMPACT ON MIXED-MODEL ASSEMBLY SYSTEMS AND SUPPLY CHAINS

by

Hui Wang

Co-Chairs: Shixin Jack Hu and Goker Aydin

Product variety has increased explosively in the past few decades and brought many challenges to manufacturing systems and supply chains. This dissertation studies the product variety induced complexity and its impact on mixed-model assembly systems and supply chains.

The first part of this dissertation develops a complexity measure for mixed-model assembly systems with different configurations, including serial, parallel and hybrid configurations. The impact of complexity on system throughput is analyzed using an approximate throughput model, which takes into consideration the operator reaction time and fatigue effect. The complexity and throughput models are then used to compare the performances of assembly systems with different configurations. Finally, a mathematical formulation is developed based on mixed-binary nonlinear programming to minimize the complexity or maximize the throughput of a mixed-model assembly system by allocating the modules to different stations.

The second part of the dissertation develops a complexity measure for assembly supply chains based on the information entropy. This complexity measure takes into account the supply chain configuration, the variety level at each node in the supply chain, and the demand ratios of the variants offered by the node. In addition, the degree of consistency between the complexity and cost is studied when the complexity and cost are used to com-
pare 1) assembly supply chains with the same configuration but different levels of product variety, and 2) assembly supply chains with the same level of product variety but different configurations.

The third part of the dissertation applies the complexity of assembly supply chains to configuration design, i.e., finding the optimal supply chain configuration with minimum complexity given the product variety level at the final assembler. The optimal supply chain configuration is studied for two special scenarios: 1) the demand share of one particular variant is bigger than that of others at the final assembler, and 2) demand shares are equal across all variants at the final assembler. In addition, a methodology is developed to find the optimal supply chain with/without assembly sequence constraints for general demands.
CHAPTER 1

INTRODUCTION

1.1 Motivation

In order to thrive in today’s market, manufacturers are motivated to provide high product variety at competitive cost. Such high product variety is enabled by modular product design, mixed-model assembly systems, and modular assembly supply chains. Modular design decomposes the product into several modules with standard interfaces and each module has a certain number of variants. High product variety can be achieved through the combinational assembly of the variants from different modules while each module still keeps the high production volume, through which the economy of scale can be maintained. For example, Herman Miller, an office-furniture manufacturer in Michigan, can produce millions of different variants for one of its office-chairs alone through assembling the variants of frame, arm rest, material, color, etc. (Herman-Miller, 2009).

The product variety has in consequence increased explosively in the past few decades. The increase of product variety has taken place in every aspect of our lives, from radio broadcast stations to milk types, from KFC menu items to contact lens types, etc. For example, the number of distinct vehicle models offered in the US rose from 44 in 1969 to 168 in 2005 (Ward’s Automotive Yearbook, 1970 & 2006). The number of styles of running shoes went from five in the early 70s to 285 in the late 90s (1998 Annual Report of the Federal Reserve Bank of Dallas). However, the increase of product variety also brings many challenges to manufacturers. Several studies have shown that high product variety has
adverse effects on manufacturing systems and supply chains, such as complicating assembly process, lowering productivity, increasing inventory cost, etc. (MacDuffie et al., 1996; Fisher and Ittner, 1999).

1.1.1 Mixed-model Assembly Systems

As product variety increased, manufacturing systems have evolved from the traditional dedicated moving assembly lines to today’s mixed-model assembly systems. Moving assembly lines invented by Henry Ford in 1913 are dedicated systems for the production of a single product with high volume (Koren, 2010). The stations of a moving assembly line are connected one after another in a serial line and a conveyor system moves partially complete products in a fixed pace from one station to another. At each station, operators only perform simple repetitive assembly tasks and the final product is completed at the end of the line. As the trend for product variety and smaller-volume-per-product emerged, mixed-model assembly lines have gradually been implemented by manufacturers to produce multiple product models in one assembly line. A mixed-model assembly line consists of a certain number of stations connected in series. Each station can process multiple products with the operator selecting an option from many variants of a module and assembling it onto the partially finished product. However, one shortcoming of mixed assembly lines is the single part flow path, i.e., the whole line breaks down if one station fails, which makes it unfit for large demand fluctuation and high product variety in today’s manufacturing. Therefore, mixed-model assembly systems, which overcome the shortcoming of serial assembly lines but still maintain the advantage of the flexibility of human operators at mixed-model assembly stations, have gained more popularity. A mixed model assembly system is composed of a certain number of mixed-model assembly stations connected in various configurations, including serial, parallel and hybrid. Notice that a mixed-model assembly line is a special mixed-model assembly system in which assembly stations are connected in a serial configuration.

However, one challenge in using mixed-model assembly systems is the increased complexity induced by product variety. As variety increases, the manufacturing process becomes quite complex. The complexity lies in production planning, part supply, to assembling pro-
cess. It is challenging to coordinate numerous small production steps that are caused by a 
large number of different variants for each module.

One possible approach to cope with this challenge is to study how product variety 
complicates the assembly process and in turn influences the performance of mixed-model 
assembly systems. Zhu et al. (2008) studied the variety induced manufacturing complexity 
in a mixed-model assembly line. Based on the various choices that the operator has to 
make at each station, Zhu et al. (2008) used the information entropy function to develop 
the station level and system level complexity models for mixed-model assembly lines. How-
ever, as discussed before, besides serial configuration, mixed-model assembly stations can 
be connected in many other configurations such as parallel and hybrid configurations, which 
result in different mixed-model assembly systems. It is well recognized that configurations 
have a profound impact on the performance of manufacturing systems (Koren et al., 1998; 
Spicer et al., 2002). Therefore, the effect of system configurations should be taken into 
consideration when studying the variety induced complexity and its impact on the perfor-
maence of mixed-model assembly systems. The first part of this dissertation is dedicated to 
the complexity model for mixed-model assembly systems with different configurations and 
studies how this variety induced complexity influences the system throughput.

1.1.2 Modular Assembly Supply Chains

Modular assembly supply chains have been used by manufacturers to handle product 
variety. In a modular assembly supply chain, the manufacturer apportions the product into 
different modules which are then outsourced and assembled by its suppliers. As a result, 
only a few pre-assembled modules will be delivered to and assembled by the final assembler, 
through which the complexity of final assembly process is reduced and the risk is shifted. 
Modular assembly supply chains have found wide applications in many industries, such as 
automotive, aerospace, electronics, etc. For instance, Volvo’s S80 model is assembled from 
23 different modules, delivered directly to the final assembly line by 17 different assembly 
units, 11 of which are operated by suppliers (Fredriksson, 2006).

Product variety also brings challenges to supply chain design. Usually the variety and 
configuration decisions are based on cost analysis. However, cost analysis for assembly
supply chains is difficult to perform for two reasons. First, sophisticated cost models for modular assembly supply chains with product variety are difficult to develop and analyze due to the network structure of an assembly supply chain and the influence of multiple products. Second, cost models require the estimation of many parameters, e.g., manufacturing costs, holding and shortage costs, transportation costs, production and transportation leadtimes, some of which are difficult to obtain in practice.

In order to overcome the above challenges in cost analysis, we introduce the complexity concept from information systems to assembly supply chains and develop a complexity measure for assembly supply chains. Unlike costs, this complexity measure does not require the estimation of any parameters other than the demand. The second part of this dissertation is dedicated to developing the complexity model for assembly supply chains in the presence of product variety and studying the relationship between the complexity and other performance measures, such as cost. One important application of the complexity measure is supply chain configuration design given the product variety level. The third part of the dissertation focuses on applying the complexity measure to the configuration design of assembly supply chains.

1.2 Research Objective and Tasks

The objective of the research is to develop models for variety induced complexity in mixed-model assembly systems and supply chains and study the impact of this complexity on the system performance. Specifically, the research includes the following three tasks.

1) To define a measure of the variety induced complexity in mixed-model assembly systems with different configurations and investigate how the complexity influences the system performance, such as throughput.

2) To define a complexity measure of assembly supply chains in the presence of product variety and study the relationship between the complexity and cost of an assembly supply chain.

3) To apply the developed complexity measure to supply chain configuration design and
develop algorithms to find the optimal configuration of assembly supply chains given
the modular product structure and level of product variety.

1.3 Organization of the Dissertation

The dissertation is presented in a multiple manuscript format. Chapters 2, 3, and 4
are written as individual research papers, including the abstract, the main body and the
references.

Chapter 2 studies the variety induced complexity in mixed-model assembly systems with
different configurations, including serial, parallel and hybrid of the two (Wang and Hu,
2010). The complexity measure takes into consideration operator choices at each station
and the system configuration. An approximate throughput model is developed for mixed-
model assembly systems by taking the complexity-based operator reaction time and fatigue
effect into consideration. Then the complexity and throughput models are used to compare
the performances of assembly systems with different configurations. It is discovered that
the complexity increases as the configuration changes from serial to hybrid and then to
parallel. The throughput decreases as fatigue effect increases from none to low and to
high. The fatigue effect on the throughput of mixed-model assembly systems with higher
complexity is higher than the effect on those with lower complexity. In a mixed-model
assembly system, the number of variants and the mix ratio of these variants do not have
to be the same for all parallel stations, which is different from traditional manufacturing
systems requiring identical and homogeneous parallel stations. Therefore, it is necessary
to investigate the number of options the operator handles at parallel stations and the
corresponding station level complexity by assigning different variants to different parallel
stations, which may in turn reduce the complexity of assembly systems and improve the
throughput. A mathematical formulation is developed based on mixed-binary nonlinear
programming to minimize the complexity or maximize the throughput of a mixed-model
assembly system through allocating the modules to different stations and assigning the
production of variants to different parallel stations.

Chapter 3 proposes a complexity measure for assembly supply chains, based on the
concept of information entropy (Wang et al., 2009). The complexity measure takes into account factors such as the supply chain configuration, the level of variety offered by each node of the supply chain, and the demand ratios across all the variants offered by a node. The relationship between the complexity and cost of an assembly supply chain is further investigated in two different scenarios. First the degree of consistency between the complexity and cost criteria is theoretically and numerically studied when they are used to compare assembly supply chains with the same configuration but different levels of product variety. Then the consistency between the complexity and cost is investigated when they are used to compare assembly supply chains with the same level of product variety, but different configurations. The results suggest that the complexity and cost criteria agree well when evaluating alternative levels of product variety that will be delivered by a given supply chain configuration, but when evaluating alternative supply chain configurations to deliver a given level of product variety, the results of complexity and cost criteria can sometimes be different.

Chapter 4 applies the complexity measure of assembly supply chains to the configuration design, i.e., finding the optimal supply chain configuration given the number of variants offered at the final assembler and the mix ratios of these variants (Wang et al., 2010). The optimal assembly supply chain configuration is studied in the following two special scenarios: 1) there is only one dominant variant among all the variants offered by the final assembler, i.e., the demand share of one particular variant is bigger than the demand share of others, and 2) demand shares are equal across all variants at the final assembler. It is shown that in the first scenario, the optimal assembly supply chain should be non-modular; but in the scenario of equal demand shares, a modular supply chain is more beneficial than a non-modular one when the product variety is high. For the scenario of general demands, a methodology is developed to find the optimal supply chain with/without assembly sequence constraints.

Finally, Chapter 5 draws the conclusions and summarizes the original contributions of the dissertation. Several topics of future research are suggested.
BIBLIOGRAPHY


CHAPTER 2

PRODUCT VARIETY INDUCED COMPLEXITY IN
MIXED-MODEL ASSEMBLY SYSTEMS AND ITS
IMPACT ON THROUGHPUT 1

ABSTRACT

A complexity measure is proposed for mixed-model assembly systems with different configurations, including serial, parallel and hybrid. The complexity measure takes into account operator choices at each station and system configurations. An approximate throughput model is developed for mixed-model assembly systems by taking into consideration operator reaction time and fatigue effect. The complexity and throughput models are used to compare the performances of assembly systems with different configurations. It is discovered that the complexity increases as the configuration changes from serial to hybrid and then to parallel. The throughput decreases as fatigue effect increases from low and to high. For the same increase of fatigue, the throughput reduction is higher in assembly systems with higher complexity than those with lower complexity. This chapter also studies how variant differentiation at parallel stations reduces the complexity and improves the throughput of mixed-model assembly systems. Traditional single-product manufacturing systems require

1Part of this chapter will appear as Wang, H. and Hu, S.J., “Manufacturing Complexity in Assembly Systems with Hybrid Configurations and Its Impact on Throughput”, Accepted by CIRP Annals-Manufacturing Technology
parallel stations to be homogeneous and identical. However, the presence of product variety in mixed-model assembly systems allows different parallel stations to produce different product variants. A mathematical formulation is developed based on mixed-binary nonlinear programming to minimize the complexity or maximize the throughput of a mixed-model assembly system through allocating the modules to different stations and assigning the production of variants to different parallel stations.

2.1 Introduction

Mass customization has been recognized as a key competitive factor for today’s manufacturers. It allows companies to provide customized products according to customers’ specific needs at near mass production cost. In mass customization, products are made from several different modules with standard interfaces and each module has a certain number of variants. High product variety is achieved through combinational assembling the variants of these modules while at the same time each module still keeps high production volume, through which the economics of scale is maintained. Such assembly is achieved using mixed-model assembly systems (MASs) as shown in Figure 2.1. In a MAS, assembly stations are connected in a certain configuration and a conveyor belt or a similar system steadily moves products from one station to another in a fixed pace. At each station, an operator chooses an option from many variants of a module and assembles it onto the partially finished product. The final product is completed at the last station. One of the challenges in using MASs is the increased complexity induced by product variety. As product variety increases, the manufacturing process becomes more complicated and the performance of MAS degrades. Several studies have shown that product variety has negative impacts on MASs, such as complicating assembly process, lowering productivity, degrading quality, etc (MacDuffie et al., 1996; Fisher and Ittner, 1999).

One possible way to handle the product variety is to design MASs that can produce multiple product models without degrading the system performance. Research conducted in this area can be divided into two categories, the long-term line balancing and the short-term product sequencing. Long-term line balancing problems focus on assigning tasks to
stations with the objective of minimizing the number of stations or the system cost at the same time satisfying the pre-determined precedence constraints. The details can be found in the review papers of Boysen et al. (2007) and Lusa (2008). Short-term scheduling focuses on the production sequence of a given number of models within the planning horizon with the objective of minimizing work overload or level part usage (Boysen et al., 2009). The effect of product variety in the research is usually modeled through either different assembly task time for different models or setup up time and cost due to switching between different models. However, the above research has not clearly answered how product variety complicates the operator’s assembly task in MASs and how this variety induced complexity influences the system performance.

In order to understand how product variety complicates the assembly process and in turn impacts the performance of MASs, some research work has been conducted in the complexity of assembly lines and manufacturing systems. First of all, we need to understand what is complexity and how to measure complexity. When one talks about complexity, s/he will first think of the things like a complicated machine composed of many interconnected parts, a few thousand lines of computer programming codes taking a long time to solve, a long string of characters without any predictable pattern, ever-changing climate, etc. One may describe something being complex or more complex than another one. But such an approach can not precisely describe what is complexity and how to compare the complexity. According to the definition of the dictionary, 2, “complexity” is defined as “the quality or state of being complex, i.e., composed of two or more parts, hard to separate, analyze, or solve”. However, this complexity definition is not sufficient for us to assess and measure complexity. Therefore, researchers in different areas propose various complexity measures.

\[\text{http://www.merriam-webster.com}\]
In the context of communication systems, Shannon (1948) defined the entropy as a measure of the uncertainty surrounding the outcome of a random experiment, which has been used by researchers as the complexity measure in many disciplines. Suh (2005) defined the complexity in the context of product design as “a measure of uncertainty in understanding what it is we want to know or in achieving a functional requirement (FR)”.

In the area of computer science, Cover and Thomas (1991) defined Kolmogorov complexity, as a measure of computational resources needed to describe a string of text.

Deshmukh et al. (1998) derived an information-theoretic entropy measure of complexity for a given combination and ratio of part types to be produced in a manufacturing system. ElMaraghy et al. (2005) proposed a code-based structural complexity index to capture the amount of information in the manufacturing systems as well as another complexity measure to represent the probability of a manufacturing systems success in delivering the desired production capacity. Zhu et al. (2008) studied the variety induced manufacturing complexity in serial, manual MASs. Based on the various choices that the operator has to make at each station, Zhu et al. (2008) used the information entropy function to develop the complexity model for both station level and system level of MASs.

One important assumption used in the above research is that all assembly stations in MASs are connected in a serial line. However, in practice there are many other configurations in which the assembly stations can be connected. For example, for a MAS with four assembly stations, there are six possible configurations, as shown in Figure 2.2. It is well recognized that configurations have a profound impact on the performance of manufacturing systems (Koren et al., 1998; Spicer et al., 2002). Therefore, it is necessary to take into account the effect of system configuration when studying the variety induced manufacturing complexity.

Figure 2.2: Different configurations for MASs with four assembly stations
complexity and its impact on the performance of MASs. In this chapter, we extend the complexity measure from a serial assembly line (Zhu et al., 2008) to MASs with different configurations. This complexity measure takes into account the factors of operators’ choices at each station and system configurations. Based on the complexity measure, we develop an approximate throughput model for MASs, which captures the likelihood of operator error during an assembly process given the product variety level and assembly cycle time. In addition, we apply the complexity and throughput models to comparing MASs with different configurations, from serial to hybrid and parallel. We finally discuss differentiating the production of variants at parallel stations to reduce the complexity of MASs and improve the throughput.

The chapter is organized as follows. In Section 2.2, we describe assembly system configurations and demand mix. Then we propose a complexity measure (Section 2.3) and develop an approximate throughput model (Section 2.4) for MASs with different configurations. In Section 2.5, we compare MASs with different configurations in terms of throughput and complexity. In Section 2.6, we discuss how to reduce the complexity and improve the throughput of MASs by differentiating the production of variants at parallel stations. Section 2.7 concludes the chapter.

2.2 Model Description

Assume that the final product consists of \( n \) modules, and for module \( i \), there are \( V_i \) different variants \( i = 1, \ldots, n \). Suppose any combinational assembly of the variants of these \( n \) modules is feasible and is counted as a distinct product variant. Therefore, the number of variants for the final product is \( N = \prod_{i=1}^{n} V_i \). Since the focus of the chapter is to study MASs with different configurations, the assembly precedence is not considered here. Suppose the customer demands for these \( N \) variants are independent and the fraction of the total demand that belongs to variant \( v \) is \( q_v \), \( v = 1, \ldots, N \), i.e., \( \sum_{v=1}^{N} q_v = 1 \). Vector \( Q = (q_1, \ldots, q_N)' \) represents the demand mix ratio for these \( N \) variants, which is also referred to the demand vector for the final product in this chapter. For example, if the product consists of four modules and each module has two variants, i.e., \( n = 4 \) and \( V_i = 2 \),
Figure 2.3: Combinational assembly of four modules and the demand vector relationships between $Q \{3,4\}$ and $Q_i = 1, 2, 3, 4$, then the total number of variants for the final product is $N = 2^4 = 16$, as shown in Figure 2.3.

Let $A \{j_1, \ldots, j_w\}$ represents the set of $w$ modules, $\{j_1, \ldots, j_w\}$, where $j_r \in \{1, \ldots, n\}$, $r = 1, \ldots, w$. Through the combinational assembly of these $w$ modules, there are $N \{j_1, \ldots, j_w\} = \prod_{r=1}^{w} V_{j_r}$ different variants. The demand vector of these $N \{j_1, \ldots, j_w\}$ variants, $Q \{j_1, \ldots, j_w\} = (q_1 \{j_1, \ldots, j_w\}, q_2 \{j_1, \ldots, j_w\}, \ldots)$, can be calculated through $Q \{j_1, \ldots, j_w\} = \Phi \{j_1, \ldots, j_w\} \times Q$, where $Q = (q_1, \ldots, q_n)'$ is the demand vector of the final product and matrix $\Phi \{j_1, \ldots, j_w\}$ is the variant selection matrix in which $\phi_{uv} = 1$ if the variant $u$ is contained in the variant $v$ of the final product and otherwise $\phi_{uv} = 0$, $u = 1, \ldots, N \{j_1, \ldots, j_w\}$, $v = 1, \ldots, N$. Equation $Q \{j_1, \ldots, j_w\} = \Phi \{j_1, \ldots, j_w\} \times Q$ can be further written in details as follows. See Figure 2.3 for the illustration.

\[
\begin{pmatrix}
q_1 \{j_1, \ldots, j_w\} \\
q_2 \{j_1, \ldots, j_w\} \\
\vdots \\
q_{N} \{j_1, \ldots, j_w\}
\end{pmatrix}_{(N \{j_1, \ldots, j_w\} \times 1)}
= 
\begin{pmatrix}
\phi_{10} & \cdots & \phi_{1N} \\
\phi_{20} & \cdots & \phi_{2N} \\
\vdots & \ddots & \vdots \\
\phi_{N0} & \cdots & \phi_{NN}
\end{pmatrix}_{(N \{j_1, \ldots, j_w\} \times N)}
\times
\begin{pmatrix}
q_1 \\
q_2 \\
\vdots \\
q_N
\end{pmatrix}_{N \times 1}
\]

(2.1)
Table 2.1: Description of serial, parallel and hybrid configurations

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Num of modules assembled at station $k$</th>
<th>Assembly cycle time of station $k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial</td>
<td>$n_k = 1$</td>
<td>$T_k = T$</td>
</tr>
<tr>
<td>Parallel</td>
<td>$n_k = n$</td>
<td>$T_k = nT$</td>
</tr>
<tr>
<td>Hybrid</td>
<td>$1 \leq n_k \leq n - 1$</td>
<td>$T_k = n_k T$</td>
</tr>
</tbody>
</table>

Assuming the same assembly time for these $n$ modules, we consider three types of configurations in this chapter, serial, parallel and hybrid. For each configuration, there are $n$ assembly stations. Suppose the designed throughput of MASs is $T$ time units/part. First let us consider a MAS with serial configuration, in which $n$ assembly stations are connected in a serial line and one module is assembled at each station. The assembly task time is the same for all stations in the serial configuration, which results in the identical cycle time of each station, $T$. Second for a MAS with parallel configuration, there are $n$ assembly stations connected in parallel and at each station, an operator assembles all these $n$ modules into the final product. Therefore, the cycle time of each station is $nT$. The third configuration we consider is hybrid configuration. A MAS with hybrid configuration has $n$ assembly stations connected in a combined configuration of serial and parallel. There are $n_k \in \{1, \ldots, n - 1\}$ modules are assembled at station $k = 1, \ldots, n$, so the cycle time of station $k$ is $n_k T$. If the number of modules assembled at station $k$ is larger than one, i.e., $n_k > 1$, then $n_k$ stations are needed to connect in parallel in order to meet the designed productivity. The details of these three configurations are summarized in Table 2.1, where $n_k$ denotes the number of modules assembled at station $k$ and $T_k$ is the cycle time of station $k$, $k = 1, \ldots, n$.

For example, suppose the product consists of four modules, which are assembled in four stations of a MAS. The MASs with serial and parallel configurations are shown in Figure 2.2 (a) and (f). In addition, there are four hybrid configurations, shown in Figure 2.2 (b) to (e). Notice that there may be more than one MAS that have the same configuration but with different modules assembled at each station. For example, Figure 2.4 (a) and (b) are two different MASs that share the same hybrid configuration but with different modules assembled at each station.

Let $A_k$ denote the set of the modules assembled at station $k = 1, \ldots, n$. Then the
Figure 2.4: Two different MASs share the same configuration. The MAS in (a) has the production of all four variants evenly distributed to two parallel stations and the MAS in (b) has variant 1 and 3 assigned to one parallel station and variant 2 and 4 assigned to another.
number of modules assembled at station $k$ is $n_k = ||A_k||$, which makes the station cycle time $T_k = n_k T$. Let $N_k$ represent the number of variants at station $k$ and $Q_k = (q_{k1}, \ldots, q_{kN_k})'$ be the demand vector of these $n_k$ variants. If $n_k > 1$, then $n_k$ parallel stations are needed in order to satisfy the designed productivity. In addition, the number of variants assembled from the modules in $A_k$ is $N^{A_k} = \prod_{j \in A_k} V_j$ and its demand vector, $Q^{A_k} = (q^{A_k}_1, q^{A_k}_2, \ldots)'$, can be calculated by Equation (2.1). Therefor, the number of variants for the operator to choose at station $k$, $N_k$, is equal to $N^{A_k} = \prod_{j \in A_k} V_j$, i.e., $N_k = N^{A_k}$, and we use $N_k$ and $N^{A_k}$ interchangeably afterwards. But the demand fraction of variant $v = 1, \ldots, N_k$ at station $k$, $q_{kv}$, is not always equal to $q^{A_k}_v$. Because if there are $n_k$ parallel stations, where these $N_k$ variants can be produced, we have the options to decide which variant and how much of that variant is produced at each parallel station. If station $k$ is one of the $n_k$ parallel stations, let $\alpha_{kv} \in [0, 1]$ denote the fraction of the variant $v = 1, \ldots, N_k$ that is assigned to produce at station $k$, i.e., $\sum_{k=1}^{n_k} \alpha_{kv} = 1$, where $\alpha_{kv} = 0$ means variant $v$ is not produced at station $k$. Then the demand fraction of variant $v$ at station $k$, $q_{kv}$, can be calculated as follows.

$$q_{kv} = \frac{\alpha_{kv} q^{A_k}_v}{\sum_{v=1}^{N_k} \alpha_{kv} q^{A_k}_v} \quad (2.2)$$

We can see that $f_k = \sum_{v=1}^{N_k} \alpha_{kv} q^{A_k}_v$ represents the production fraction assigned to station $k$, where station $k$ is one of the $n_k$ parallel stations. If the production of all variants is evenly distributed to all parallel stations, i.e., $\alpha_{kv} = 1/n_k$, $f_k = 1/n_k$, then we have $q_{kv} = q^{A_k}_v$. However, if we set $\alpha_{kv} = 0$ for some variants at station $k$, we can differentiate the production of variants at different parallel stations and therefore reduce the number of variant choices for the operator at some stations. See Figure 2.4 for the illustration. The MAS in Figure 2.4 (a) has the production of all four variants evenly distributed to two parallel stations and the MAS in Figure 2.4 (b) has variant 1 and 3 assigned to one parallel station and variant 2 and 4 assigned to another.
2.3 Complexity of MASs with Different Configurations

In this section, based on Shannon’s information entropy we define complexity measures for MASs with serial, parallel and hybrid configurations. The concept of information entropy was originated from the context of communication systems and used to measure the uncertainty of outcomes in a random experiment (Shannon, 1948). Suppose a random experiment has $M$ possible outcomes, whose probabilities of occurrence are $p_1, \ldots, p_M$. The information entropy of the random experiment, defined as $H = -\sum_{m=1}^{M} p_m \log_2 p_m$, is used to measure how uncertain we are about the outcome. Zhu et al. (2008) used the entropy function to define the complexity of an assembly station in MASs as follows, where there are $M$ different variants the operator can choose and demand fraction of variant $m$ is $p_m$, $m = 1, \ldots, M$.

$$H_k = -\sum_{m=1}^{M} p_m \log_2 p_m$$  \hspace{1cm} (2.3)

The complexity defined in Equation (2.3) tells us the uncertainty level of the next variant the operator will choose at the station. In this chapter, we extend this complexity definition from one assembly station to the whole assembly system and define complexity measures for MASs with various configurations.

2.3.1 Complexity of a MAS with Serial Configuration

In a MAS with serial configuration, there are $n$ serially connected assembly stations and at each station an operator assembles one module. Without loss of generality, we assume module $i$ is assembled at station $i$, $i = 1, \ldots, n$, i.e., $A_i = \{i\}$ and $N^{A_i} = V_i$. We define the complexity of a MAS with serial configuration as the sum of complexity of each station.

$$H_{Serial} = \sum_{i=1}^{n} H_i = -\sum_{i=1}^{n} \sum_{v=1}^{V_i} q_{iv} \log_2 q_{iv}$$  \hspace{1cm} (2.4)

This complexity definition can be interpreted through the following random experiment. Suppose we randomly select one variant from each station, i.e., $v_i \in \{1, \ldots, V_i\}, i = 1, \ldots, n$, and assemble them into the final product. In addition, the probability of selecting a certain variant, $v_i$, at station $i$ is equal to its demand share, $q_{iv}$. The information entropy of this
random experiment is equal to Equation (2.4) and defined as the complexity of MASs with serial configuration. Generally speaking, this complexity definition indicates the uncertainty level of the next final product that will be assembled.

2.3.2 Complexity of a MAS with Parallel Configuration

In a MAS with parallel configuration, there are \( n \) assembly stations connected in parallel and an operator assembles all the \( n \) modules at one station, i.e., \( A_i = \{1, 2, \ldots, n\} \) and \( N^{A_i} = N = \prod_{j=1}^{n} V_j, \ i = 1, \ldots, n \). We define the complexity of MASs with parallel configuration as follows, where \( f_i = \sum_{v=1}^{N} \alpha_{iv} q_{v}^{A_i} \) is the production fraction assigned to station \( i = 1, \ldots, n \) and \( q_{iv} \) is the demand fraction of variant \( v \) at station \( i \).

\[
H_{\text{Parallel}} = -\sum_{i=1}^{n} \sum_{v=1}^{N} f_i q_{iv} \log_2 f_i q_{iv} = -\sum_{i=1}^{n} f_i \sum_{v=1}^{N} q_{iv} \log_2 q_{iv} - \sum_{i=1}^{n} f_i \log_2 f_i
\]

This complexity definition can be interpreted through the following random experiment. Suppose we first randomly choose a station \( i, i \in \{1, \ldots, n\} \), and then select one variant \( v_i \) from that station, i.e., \( v_i \in \{1, \ldots, V_i\} \). And the probability of choosing station \( i \) is equal to the production fraction of the station, \( f_i \), and the probability of selecting variant \( v_i \), is equal to its demand share at that station, \( q_{iv} \). Since the probability of choosing station \( i \) and then selecting variant \( v_i \) at station \( i \) is \( f_i \cdot q_{iv} \), the information entropy of this random experiment is equal to Equation (2.5). Here we use this information entropy as the complexity measure of MASs with parallel configuration. Generally speaking, this complexity definition indicates the uncertainty level of what the next final product will be demanded and where it will be assembled.

Notice that the complexity of MASs with parallel configuration defined in Equation (2.5) can be divided into two parts. The first part, \( \sum_{i=1}^{n} f_i H_i \), is the sum of complexity of each parallel station weighted by its demand fraction, \( f_i \). The second part, \( -\sum_{i=1}^{n} f_i \log_2 f_i \), is the complexity caused by splitting the production into different parallel stations. If the production is evenly distributed to each parallel station, i.e., \( f_i = 1/n \) and \( q_{iv} = q_v \), then
we can rewrite Equation (2.5) in a simplified form as follows.

\[ H_{Parallel} = - \sum_{v=1}^{N} q_v \log_2 q_v + \log_2 n \]  

(2.6)

2.3.3 Complexity of a MAS with Hybrid Configuration

A MAS with hybrid configuration can be decomposed into smaller-size MASs with serial and parallel configurations. Based on the complexity definition of serial and parallel MASs shown in Equation (2.4) and (2.5), we can use an iterative aggregation algorithm to calculate the complexity of a MAS with hybrid configuration. For example, Figure 2.5 demonstrates the procedure of using the aggregation algorithm to calculate the complexity of a MAS that have eight stations connected in a hybrid configuration. The algorithm starts from a MAS shown in (I). When a serial or parallel configuration is met, Equation (2.4) or (2.5) is used to calculate the complexity and then that serial or parallel configuration is treated as one assembly station with the calculated complexity value. Repeat this process until only one serial or parallel configuration remains.
2.4 Throughput Model of MASs with Different Configurations

In this section, we first build the reliability model for an assembly station based on the operator’s reaction time and fatigue effect. Then based on the station reliability model, we develop the throughput model for MASs with different configurations.

2.4.1 Reliability Model for a Mixed-model Assembly Station

In this chapter, the reliability for an assembly station refers to the probability that the operator of the station chooses the right option and successfully assembles it onto the product within the given cycle time. Since we are interested in how product variety complicates the manual assembly process and in turn influences the operator’s performance, it is assumed that once the operator chooses the right variant, s/he will be able to successfully assemble it without any mistake. Yang et al. (1997) performed empirical study on the reliability of operators’ diagnosis and decision-making process based on human cognitive reliability (HCR) model. Their study suggested that it was reasonable to use Weibull or Lognormal distributions to fit HCR data. Hence in this section, we use Weibull distribution to develop the reliability model for a station in MASs based on the operator reaction time and human fatigue effect.

Based on the model in Section 2.2, the cycle time of station \( k = 1, \ldots, n \), is \( T_k = n_k T \), where \( n_k \) is the number of modules assembled at station \( k \) and \( T \) is the designed throughput of MASs. Within the cycle time \( T_k \), the operator needs to finish the following two tasks: choosing the right variant and then assembling it onto the partially finished product. Let \( T_k^S \) denote the average time needed to choose the right variant and \( T_k^A \) be the time that is needed to finish the assembly task at station \( k \). Hyman (1953) studied the performance of human selection activities by measuring average reaction times, i.e., how quickly a person can make a choice in response to a stimulus. He found out that the average reaction time is approximately a linear function of information entropy conveyed by the stimulus. In station \( k = 1, \ldots, n \), the information entropy conveyed by stimulus is equal to the complexity of the station, \( H_k \). Therefore, we can write the average selection time as \( T_k^S = a + bH_k \), where \( a \)
and \( b \) are two ergonomics constants if all operators are assumed to be homogeneous. Since we assume the task time of assembling all modules is the same, assembly time at station \( k \) can be written as \( T_k^A = n_k d \), where \( d \) is the time needed to assemble one module.

The reliability of station \( k \) is impacted by the following two factors. The first one is within one cycle time \( T_k \), the difference in the time available to make choices, \( T_k - T_k^A \), and the time needed to make choices, \( T_k^S \). This can be measured by ratio \( \frac{bH_k}{T_k - T_k^A} \). Bigger \( \frac{bH_k}{T_k - T_k^A} \) means less time available for the operator to make the choice, which results in lower reliability.

The second factor is the fatigue effect, which plays an important role in the human performance. There are two types of fatigue that could influence the performance of an operator, physical fatigue and mental fatigue. Physically fatigue refers to the discomfort that occurs in the body as a result of repeated and sustained exertions (Bystrom and Fransson-Hall, 1994). Mental fatigue refers to a psychobiological state caused by prolonged periods of demanding cognitive activity (Marcora et al., 2009). Physical fatigue is usually measured by percentage of maximum voluntary contraction (\%MVC) in one cycle, where \%MVC is defined as \((\text{Force of exertion})/ (\text{Operator strength capacity})\). For example, if a worker with the grip strength of 120 pounds picks up a 40-pound toolbox, \%MVC he would exert is \( \frac{40}{120} = 33\% \). Bystrom and Fransson-Hall (1994) discovered that unacceptable physical fatigue occurred when the average workload, \%MVC\(_{\text{avg}}\), exceeded 17\% for combinations of work-rest cycles in intermittent work. Consider the MAS with serial configuration where the operator assembles one module at one station within one cycle time \( T \). The operator incurs \%MCV\(_W\) during the assembly time, \( d \), and \%MCV\(_R\) = 0 during the rest time, \( T - d \). The average work load in one cycle is

\[
\%MVC_{\text{avg}} = \frac{\%MCV_W \cdot d + \%MCV_R \cdot (T - d)}{T} = \%MCV_W \cdot \frac{d}{T}
\]

which is assumed less than 17\%, i.e., \%MVC\(_{\text{avg}} < 17\%\). Then we can calculate the average workload at station \( k = 1, \ldots, n \), \%MVC\(_{\text{avg}}\), as follows, where assembly task time is \( n_k d \)
and cycle time is $n_k T$.

$$\%MVC_{k,avg} = \frac{\%MCV_W \cdot n_k d + \%MCV_R \cdot n_k (T - d)}{n_k T} = \%MCV_W \cdot \frac{d}{T} = \%MVC_{avg} < 17\%$$

From the above calculation, we conclude that average work load in one cycle at station $k = 1, \ldots, n$ is less than 17%, so no physical fatigue will occur at any station of MASs. Hence we will only consider mental fatigue here. According to the definition, the mental fatigue at station $k = 1, \ldots, n$, depends on the time of demanding cognitive activity at the station, which is the selection time, $T_k^S = a + b H_k$. In this chapter, an effectiveness index $\eta(H_k)$ is used to represent the effect of fatigue on the performance of the operator at station $k = 1, \ldots, n$, where $\eta(x)$ is a decreasing function of $x$. Higher $H_k$ means higher mental fatigue effect, which results in lower efficiency of the operator.

Based on above two factors, we define the reliability of station $k = 1, \ldots, n$ as follows, where $\beta > 1, \eta(H_k) \neq 0$.

$$R_k = e^{-\left(\frac{b H_k}{(T_k - T_k^m \eta(H_k))}\right)^\beta} \quad (2.7)$$

The reliability function in Equation (2.7) is a Weibull distribution with increasing hazard rate ($\beta > 1$). The effectiveness index function $\eta(H_k)$, a decreasing function of $H_k$, is the scale parameter of the Weibull distribution, which determines the level of spread out for the distribution. For example, $\eta(H_k) = \frac{1}{H_k}$ is one possible effectiveness index function. Basically this effectiveness index function tells us that for two assembly stations that have the same $\frac{b H_k}{(T_k - T_k^m \eta(H_k))}$ ratio, the one with lower effectiveness index has lower reliability. In addition, a few properties of the reliability for one assembly station are noticed. (1) When there is no variant selection for the operator, i.e., $H_k = 0$, there is no occurrence of error and stations never break down, i.e., $R_k = 1$. (2) For two assembly stations with the same ratio of $\frac{b H_k}{T_k - T_k^m}$, the one with higher $H_k$ has lower reliability because of longer cognitive activity and higher mental fatigue effect.
2.4.2 Throughput Model for MASs

Based on the reliability model of one assembly station, shown in Equation (2.7), the throughput of MASs can be calculated using the performance analysis methodologies of manufacturing systems developed in Koren et al. (1998). Suppose a manufacturing system is composed of \( n \) stations and the reliability of station \( k \) is \( R_k \), \( k = 1, \ldots, n \). Let \( X_k \in \{0, 1\} \) represent the state of station \( k = 1, \ldots, n \), where 0 stands for ‘failure’ and 1 stands for ‘working’, and \( W(X_1, \ldots, X_n) \) is the throughput when the system state is \( (X_1, \ldots, X_n) \). Then the expected throughput of this manufacturing system can be calculated as follows.

\[
TH = \sum_{X_1=0}^{1} \cdots \sum_{X_n=0}^{1} R_{X_1} (1 - R_1)^{1-X_1} \cdots R_{X_n} (1 - R_n)^{1-X_n} \cdot W(X_1, \ldots, X_n)
\] (2.8)

2.5 Comparison of MASs with Different Configurations

In this section, we apply the complexity and throughput models to comparing MASs with different configurations. We first describe the comparison methodology we use in this chapter, based on which numerical studies are then conducted. In order to focus on configuration comparison, here we assume that the production of variants is evenly assigned to all parallel stations. Therefore, the number of variants and the demand vector of these variants are the same at all parallel stations, i.e., \( f_k = 1/n_k \), \( \alpha_{kv} = 1/n_k \), where station \( k = 1, \ldots, n_k \) is one of \( n_k \) parallel stations and variant \( v = 1, \ldots, N_k \) is the one of the \( N_k = \prod_{i \in A_k} V_i \) variants at station \( k \).

Given that the final product consists of \( n \) modules, which are assembled in \( n \) stations of MASs, the configuration comparison is divided into the following two steps.

Step 1: Generate all system configurations composed of \( n \) stations. Webbink and Hu (2005) developed a distribution algorithm that can quickly generate the system configurations with \( n \) assembly stations. The configuration is represented as a character string from the set \{1, (, )\}, where “1” represents a station and a parenthesis represents a switching between serial and parallel distribution. Figure 2.6 (a) illustrates how to use the distribution algorithm to generate the system configurations of four stations. There are 10 generated configurations, which can be divided into two categories, symmetric configurations (I, II, IV,
V, VI, X) and asymmetric configurations (III, VII, VIII, IX). A symmetric configuration is defined as a configuration, in which a symmetric line can be found and all parallel stations are homogeneous stations with the same cycle time and function (Ko and Hu, 2008). As the opposite, parallel stations in an asymmetric configuration are not always homogeneous. In practice, symmetric configurations have found to be more efficient in the system design and control, so they are the commonly used configurations in most of the literatures. Therefore, in this chapter, we only consider the MASs of symmetric configurations, which are the six configurations shown in Figure 2.6 (b).

**Step 2:** Within each configuration, find the MAS with minimum complexity and the MAS with maximum throughput. As discussed in Section 2.2, sometimes there are more than one MAS within one configuration due to the different assembly locations of modules. In order to compare the different configurations in terms of complexity and throughput, we need to compare all MASs within one configuration and choose the one with minimum complexity and the one with maximum throughput.

Figure 2.6: Distribution algorithm to generate system configurations of four stations, where * stands for the asymmetric configurations. Among these 10 configurations, six are symmetric and four are asymmetric.
Given the assumption that the production of all variants is evenly distributed to all parallel stations, we can write the complexity of MASs as follows based on Equation (2.4) and (2.6), where $q_v$ is the demand share of variant $v = 1, \ldots, N$ for the final product and function $g^S$ is the complexity function determined by the configuration.

$$H = -\sum_{v=1}^{N} q_v \log_2 q_v + g^S \tag{2.9}$$

Based on Equation (2.9), we can see that all MASs within each configuration have the same complexity value, so we do not need to enumerate all MASs to find the one with minimum complexity. However, MASs within each configuration do not always have the same throughput based on Equation (2.8). Therefore, in order to find the MAS with maximum throughput, we need to enumerate all MASs within each configuration and compare their throughput. The configuration comparison is then conducted based on the MAS with maximum throughput of each configuration.

Next, we perform the numerical studies based on the same example used in Section 2.2, in which the product consists of four modules, i.e., $n = 4$ and each module has two variants, i.e., $V_i = 2$, $i = 1, \ldots, 4$. The total number of variants for the final product is $N = 2^4 = 16$, as shown in Figure 2.3. There are six symmetric configurations that we will compare in terms of complexity and throughput, including one serial, one parallel and four hybrids, as shown in Figure 2.6 (b).

**Case 1: Evenly distributed demands**

Our first numerical study assumes that the demand fraction for these 16 product variants
are the same, i.e., $q_v = \frac{1}{16}, v = 1, \ldots, 16$. Therefore, the demand vector for module $i$ is $Q^{(i)} = (0.5, 0.5)$ and the information entropy of module $i$’s demand is $H^{(i)} = 1, i = 1, \ldots, 4$.

The number of variants at station $k$ is $N_k = 2n_k, k = 1, \ldots, 4$, and the demand share of variant $v = 1, \ldots, N_k$ is $q_{kv} = \frac{1}{N_k} = \frac{1}{2n_k}$, where $n_k$ is the number of modules assembled at station $k$. We calculate the complexity of MASs in these six configurations by Equation (2.9) and the results are shown in Figure 2.7.

From Figure 2.7, we can see that the complexity increases as the configuration changes from serial to hybrid and then to parallel. This is because the complexity of MASs is the sum of complexity of the products and the complexity of the configuration, as shown in Equation (2.9). All six configurations here have the same demand vector for final products, as such the complexity ranking of MASs here is determined by the complexity of the configuration.

Based on Equation (2.3), the complexity of station $k = 1, \ldots, 4$ is equal to the number of modules assembled at the station, i.e., $H_k = n_k$, which is not affected by the specific modules assigned to the stations any more. Therefore, the MASs within one configuration also have the same throughput because the throughput of MASs is a function only determined by $H_k$. Therefore we do not need to enumerate and compare all MASs within one configuration in terms of throughput.

The following parameters are used here for the throughput analysis. The cycle time for the MAS with serial configuration, configuration (a) in Figure 2.6 (b), is $T = 60$ sec, which makes the system throughput 1 part/min. Assume the time needed to assemble one module is $d = 50$ sec. The effectiveness index function takes the form, $\eta(H) = \frac{1}{(2H)^\delta}, \delta \geq 0$. By changing the value of $\delta$, we can change the influence of fatigue. When $\delta = 0$, it means no fatigue effect exists and the operator performance is not affected by the time of cognitive activity, i.e, $\eta(H) = 1$. When $\delta$ increases, the effect of fatigue increases and the operator performance goes down. Assume that $b = 2$ and $\beta = 2$. Finally based on the complexity of each station, we calculate the reliability of each station by Equation (2.7) and the throughput of the six configurations by Equation (2.8). Figure 2.8 shows the throughput of the six configurations as a function of $\delta$.

The result of throughput shown in Figure 2.8 is not as straightforward as complexity. In the case of no fatigue effect, i.e., $\delta = 0$, the result is the same as the traditional manufac-
Figure 2.8: Throughput of the six configurations as a function of $\delta$ under evenly distributed demands.

turing systems, where the configuration with higher number of parallel stations has higher throughput, i.e., $TH^{Parallel} > TH^{Hybrid} > TH^{Serial}$. For each configuration, as fatigue effect $\delta$ increases, the throughput decreases. This is because a higher fatigue effect reduces the reliability of each station and in turn reduces the throughput of MASs. For different configurations, the fatigue effect $\delta$ has more influence on the configurations that have higher complexity. For example, when $\delta$ changes from 0 to 0.65, the throughput of the parallel configuration reduces 0.411 part/min while the throughput of the serial configuration only reduces 0.178 part/min.

As the fatigue effect $\delta$ increases, the optimal configuration in terms of throughput moves from parallel to hybrid then to serial. This can be explained as follows: when the fatigue effect is low, the reduction of station reliability in all configurations is not significantly different from each other, which makes the result similar to the case of no fatigue effect. When fatigue effect increases, station level reliability reduces more significantly in the configurations with higher number of parallel stations, which results in more throughput loss in the parallel configuration. The hybrid configuration that has the advantage of parallel structures and incurs medium level reduction of station reliability becomes the optimal configuration. When the fatigue effect $\delta$ is high, the shortcoming of serial structure is offset by the advantage of low complexity at each station, which brings high reliability at each station of the serial configuration.
Case 2: Generally distributed demands

Our second numerical study uses the general demand vector for the final product. For example, we choose the following demand vector for the final product, \( Q = (0.00075, 0.00075, 0.003, 0.003, 0.00425, 0.00425, 0.017, 0.017, 0.01425, 0.01425, 0.057, 0.057, 0.08075, 0.08075, 0.323, 0.323) \), which results in the following demand vectors for each module, \( Q^{(1)} = (0.05, 0.95) \), \( Q^{(2)} = (0.15, 0.85) \), \( Q^{(3)} = (0.2, 0.8) \) and \( Q^{(4)} = (0.5, 0.5) \). Based on the demand vector of these four modules, we can calculate the information entropy of each module’s demand as follows, \( H^{(1)} = 0.286 \), \( H^{(2)} = 0.610 \), \( H^{(3)} = 0.722 \) and \( H^{(4)} = 1 \), which also indicates the demand uncertainty level for each module.

We first calculate the complexity of MASs in the six configurations based on Equation (2.9), and the result is shown in Figure 2.9. We can see that the result of comparison is the same as the case of evenly distributed demands, in which complexity increases as the configuration changes from serial to hybrid and then to parallel. This is because the complexity of MASs can be written as the sum of the complexity of the final product and the complexity of the configuration, based on Equation (2.9). Changing the demand vector for the final product will only make the first part, complexity of the final product, changes the same amount for all configurations, which makes the comparison result still determined by the complexity of the configurations.

Notice that the MASs within one configuration do not have the same throughput here. Therefore in order to compare different configurations in terms of throughput, we enumer-
Figure 2.10: Throughput of the six configurations as a function of δ under generally distributed demands

ate and compare all MASs within each configuration and find the MAS with maximum throughput and then compare them. The comparison result is summarized in Figure 2.10.

We can see that the result is similar to the case of evenly distributed demands. Within each configuration, throughput decreases as fatigue effect δ increases and fatigue effect δ has more impact on the configurations with higher complexity. As fatigue effect δ increases, the optimal configuration moves from parallel to hybrid then to serial. However, a few differences are also noticed here. First, the fatigue effect threshold through which the optimal configuration moves from hybrid to serial here is much bigger than the case of evenly distributed demands, i.e., δ_{General} ≈ 1.8 ≫ δ_{Even} ≈ 0.60. It is because the demand vector for the final product is less evenly distributed here, which results in lower complexity at stations and higher station level reliability for the same configuration. Therefore, the benefit of low complexity at serial stations needs bigger δ to surpass the disadvantage of the serial configuration. In addition, when the optimal configuration moves from parallel to hybrid, the specific hybrid configurations chosen in these two studies are different, where configuration (b) is the optimal hybrid configuration for generally distributed demands while configuration (d) is the optimal one for evenly distributed demands. It is mainly because the benefits of different hybrid configurations varies and usually are determined by the demand vector of the final product, which are different in these two studies. Therefore,
two different demand vectors maybe will favor different hybrid configurations in order to maximize the throughput.

Besides the comparison of MASs in different configurations, here we also look at the throughput of MASs within the same configuration. For example, there are six different MASs that share configuration (b), as shown in Figure 2.11. We study the throughput of these six MASs as the functions of fatigue effect $\delta$, and the result is shown in Figure 2.12. The similar results are also found in MASs within other configurations. When fatigue effect $\delta$ is small, the throughput of these six MASs is very close to each other. However, as $\delta$ increases, the difference increases. It is because when $\delta$ is small, the reliability of each station is high and the difference is small, which makes all MASs within one configuration have
the similar throughput. As \( \delta \) increases, the reliability difference between stations increases, which increases throughput differences of these six MASs. Another thing we notice is that in the MASs with the highest throughput under different \( \delta \) value, module 4 is always assigned to parallel stations. It agrees with the design principles of manufacturing systems, in which the machine with lowest reliability should be put in the parallel stations. Because module 4 has the highest information entropy for its demand, then the station where module 4 is assembled will have the lowest throughput and should be configured in parallel.

### 2.6 Product Variant Differentiation in MASs

In Section 2.5, it is assumed that the production of each variant is evenly assigned to all parallel stations. However, as the example shown in Figure 2.4, through assigning different variants to different parallel stations, the number of variants at some parallel stations can be reduced. Figure 2.13 is another example to illustrate using differentiating production of variants at different parallel station to reduce the complexity of MASs and increase the throughput.

As shown in Figure 2.13, the product consists of two modules which are assembled at two parallel stations of a MAS. Suppose each module has two variants and the final product has four different variants with mix ratio \( Q = (1/2,0,0,1/2) \). Figure 2.13 (a) is the MAS with production evenly distributed and Figure 2.13 (b) is the MAS using variant differentiation, in which variant 1 is assigned to one station and variant 4 is assigned to another station.
Table 2.2: An example to illustrate using differentiating the production of variants to reduce the complexity of MASs and increase the throughput

<table>
<thead>
<tr>
<th></th>
<th>MAS without variant differentiation</th>
<th>MAS with variant differentiation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_{11} = 0.5$</td>
<td>$\alpha_{21} = 0.5$</td>
<td>$\alpha_{11} = 1$</td>
</tr>
<tr>
<td>$\alpha_{12} = 0.5$</td>
<td>$\alpha_{22} = 0.5$</td>
<td>$\alpha_{12} = 0$</td>
</tr>
</tbody>
</table>

| Complexity          | 2                                   | 1                               |
| Throughput          | $TH \leq 1$                         | 1                               |

The complexity and throughput of these two MASs are calculated based on Equation (2.5) and (2.8) in which the same parameters used in Section 2.5 are also used here. The results are summarized in Table 2.2.

From this example, we can see that through properly setting the production fraction of variants at different parallel stations, $\alpha_{kv}$, we can reduce the complexity and improve the throughput of MASs. In this section, we study how to use product variant differentiation to reduce the complexity of MASs and improve the system throughput for a given configuration. First, based on mixed-binary nonlinear programming (MBNP), we develop the mathematical formulation to find the MAS with minimum complexity and the one with maximum throughput for a given configuration. Then we perform numerical studies based on the developed model.

For explanation convenience, we divide the $n$ assembly stations into $m + 1$ disjoint sets as follows. Let set $\{1, \ldots, l\}$ represents the single serial stations in which only one module is assembled and set $P^1 = \{l + 1, \ldots, l + s_1\}$ represents the group of $s_1$ parallel stations at each of which $s_1$ modules are assembled. Similarly, set $P^t = \{l + s_1 + \ldots + s_{t-1} + 1, \ldots, l + s_1 + \ldots + s_{t-1} + s_t\}$ represents the $t^{th}$ group of $s_t$ parallel stations at each of which $s_t$ modules are assembled, $t = 2, \ldots, m$. Therefore, we can write $n = l + \sum_{t=1}^{m} s_t$.

The following mathematical programming formulation can be used to find the MAS with minimum complexity and the MAS with maximum throughput within one configuration.

Minimize $\sum_{i=1}^{l} H_i + g(H_{l+1}, f_{l+1}, \ldots, H_n, f_n)$  \hspace{1cm} (2.10)

or Maximize $\sum_{X_1=0}^{1} \sum_{X_n=0}^{1} R_1^{X_1}(1 - R_1)^{1-X_1} \cdots R_n^{X_n}(1 - R_n)^{1-X_n} \cdot W(X_1, \ldots, X_n)$  \hspace{1cm} (2.11)
Subject to:

$$N_i = \prod_{j=1}^{n} V_j^{X_{ij}}, \quad i = 1, \ldots, l \quad (2.12)$$

$$Q_i = (q_{i1}, \ldots, q_{iN_i})' = \sum_{j=1}^{n} X_{ij} \cdot (\Phi^{[j]} \times Q), \quad i = 1, \ldots, l \quad (2.13)$$

$$\tilde{N}_t = \prod_{j=1}^{n} V_j^{Y_{tj}}, \quad t = 1, \ldots, m \quad (2.14)$$

$$P_t = (p_{t1}, \ldots, p_{tN_t})' = \sum_{j=1}^{n} \sum_{j_{1t} = 1}^{n} (Y_{tj_1} \cdots Y_{tj_{nt}}) \cdot (\Phi^{[j_1 \cdots j_{nt}]} \times Q), \quad t = 1, \ldots, m \quad (2.15)$$

$$N_i = \tilde{N}_t, \quad i = l + s_0 + \ldots + s_{t-1} + k, k = 1, \ldots, s_t \quad (2.16)$$

$$\tilde{Q}_i = (\tilde{q}_{i1}, \ldots, \tilde{q}_{iN_i})' = (p_{t1} \cdot \alpha_{i1}, \ldots, p_{tN_i} \cdot \alpha_{iN_i})', i = l + s_0 + \ldots + s_{t-1} + k, k = 1, \ldots, s_t \quad (2.17)$$

$$f_i = \sum_{v=1}^{N_i} q_{iv}, \quad i = l + s_0 + \ldots + s_{t-1} + k, k = 1, \ldots, s_t \quad (2.18)$$

$$Q_i = (q_{i1}, \ldots, q_{iN_i})' = (\frac{\tilde{q}_{i1}}{f_i}, \ldots, \frac{\tilde{q}_{iN_i}}{f_i})', \quad i = l + s_0 + \ldots + s_{t-1} + k, k = 1, \ldots, s_t \quad (2.19)$$

$$H_i = \sum_{v=1}^{N_i} q_{iv} \log_2 q_{iv}, \quad i = 1, \ldots, n \quad (2.20)$$

$$R_i = e^{-\left(\frac{b H_i}{(T_i - T_{ti}) \theta(\mu_i)}\right)^\beta}, \quad i = 1, \ldots, n \quad (2.21)$$

$$\sum_{i=1}^{l} X_{ij} + \sum_{t=1}^{m} Y_{tj} = 1, \quad j = 1, \ldots, n \quad (2.22)$$

$$\sum_{j=1}^{n} X_{ij} = 1, \quad i = 1, \ldots, l \quad (2.23)$$

$$\sum_{j=1}^{m} Y_{tj} = s_t, \quad t = 1, \ldots, m \quad (2.24)$$

$$\sum_{k=1}^{s_t} \alpha_{i+k,v} = 1, \quad i = l + s_0 + \ldots + s_{t-1}, v = 1, \ldots, N_i \quad (2.25)$$

$$X_{ij}, Y_{tj} \in \{0, 1\}, \quad i = 1, \ldots, l, \quad t = 1, \ldots, m, \quad j = 1, \ldots, n \quad (2.26)$$

$$0 \leq \alpha_{iv} \leq 1, \quad i = l + 1, \ldots, n, v = 1, \ldots, N_i \quad (2.27)$$
Basically it is a resource allocation problem with the objective function of minimizing the complexity or maximizing the throughput. Binary variables $X_{ij}$ and $Y_{tj}$ are used to represent whether module $j = 1, \ldots, n$ is assigned to station $i = 1, \ldots, l$ or the $t^{th}$ parallel station group, $t = 1, \ldots, m$, or not. Variables $\alpha_{iv}$ represents the production fraction of variant $v = 1, \ldots, N_i$ at parallel station $i = l + 1, \ldots, n$. Constraints (2.12) and (2.13) are used to calculate the number of variants and the demand vectors of station $i = 1, \ldots, l$. Constraints (2.14), (2.15), (2.16), (2.17), (2.18), and (2.19) are used to calculate the number of variants and the demand vector at each parallel station taking product variant differentiation into consideration. Constraint (2.20) calculates the complexity of each station and constraint (2.21) calculates the reliability of each station based on the station level complexity. Constraint (2.22) makes sure only one module is assigned to one serial station or one parallel station group. Constraints (2.23) and (2.24) guarantee the number of modules at station $i = 1, \ldots, l$ is one and the number of modules assigned to $t^{th}$ parallel station group, $t = 1, \ldots, m$, is $s_t$. Constraint (2.25) makes sure the sum of production fraction at parallel stations is one. Constraint (2.26) and (2.27) requires decision variable $X_{ij}$ and $Y_{tj}$ to be binary integer and $\alpha_{iv}$ between 0 and 1.

Next, we conduct the numerical studies based on the developed mathematical formulation. The same example used in Section 2.2 and Section 2.5 is also used here, in which the product consists of four modules and each module has two variants, i.e., $n = 4, V_i = 2, i = 1, \ldots, 4$. The total number of variants for the final product is $N = 2^4 = 16$ and there are six possible configurations for MASs shown in Figure 2.6 (b). The parameters used in Section 2.5 are also used here to analyze the throughput, where the designed throughput is $T = 60$ sec/part and the time needed to assemble one module is $d = 50$ sec and the effectiveness index function takes the form, $\eta(H) = \frac{1}{(\delta H)^\gamma}$ with fatigue effect $\delta = 0.25$. Suppose configuration (b) has been chosen and next we study how to use the developed mathematical formulation to find the MAS with minimum complexity and the MAS with maximum throughput within configuration (b).

We divide the four stations of configuration (b) into two groups, where set $\{1, 2\}$ is the group of single serial stations and $P^1 = \{3, 4\}$ represents the group of two parallel stations with two modules are assembled at each of them, i.e., $l = 2, m = 1, s_1 = 2$. Given
configuration (b), objective functions (2.10) and (2.11) in the mathematical formulation above can be written in details as follows.

\[
\text{Minimize } \sum_{i=1}^{2} H_i + \sum_{i=3}^{4} f_i H_i - f_3 \log_2 f_3 - f_4 \log_2 f_4 \tag{2.28}
\]

or Maximize \( R_1 R_2 \max(f_3, 0.5) R_3 (1 - R_4) + \max(f_4, 0.5) R_4 (1 - R_3) \]

\[+ (\max(f_3, 0.5) + \max(f_4, 0.5)) R_3 R_4 \] \tag{2.29}

The numerical study is conducted based on four different final demand vectors, as shown in Table 2.3. For the first three demand vectors, we assume the demand share of the existent variants are evenly split and the demand share of non-existing variants are zero. We start from eight variants and then increase the number variants to 12 and 16. The last demand vector is randomly chosen. The Excel Solver is chosen here to solve the problem formulated by mixed-binary nonlinear programming shown in Equation (2.28) and (2.29). The results are summarized in Table 2.3, which includes assembly location of each module and the production splitting of the variants assembled at two parallel stations, station 3 and 4.

From Table 2.3, we can see that differentiating the production of variants at parallel station helps to reduce the complexity and increase the throughput of MASs. For the complexity, it is always more beneficial to completely split the variants into different parallel stations. Because the complexity of a MAS is the sum of weighted complexity of each station and the complexity of the configuration. Completely splitting the production of variants can reduce the station level complexity and this uneven splitting also makes the complexity of configuration reduced as well. For the throughput analysis, it is more complicated and sometimes completely splitting is not optimal. For the first demand vector, there are only two variants from assembling the variants of module 1 and 2. Therefore, splitting these two variants into two parallel stations separately can make the complexity at station 3 and 4 as zero, i.e., \( H_3 = H_4 = 0 \), which will result in minimum complexity and maximum throughput. In the second case where the number of variants increases to 12, the number of variants from assembling module 1 and 2 also increases from two to three. So splitting these variants into parallel stations can not make station level complexity zero anymore. In
Table 2.3: Numerical study results of using differentiating the production of variants to reduce the complexity of MASs and increase the throughput

<table>
<thead>
<tr>
<th>Q = (1/8, 1/8, 1/8, 1/8, 0, 0, 0, 0, 1/8, 1/8, 1/8, 1/8)</th>
<th>Minimizing Complexity</th>
<th>Maximizing Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q = (1/8, 1/8, 1/8, 1/8, 0, 0, 0, 0, 1/8, 1/8, 1/8, 1/8)</td>
<td>Station 1 M3 Station 1 M3</td>
<td></td>
</tr>
<tr>
<td>Q = (1/12, 1/12, 1/12, 1/12, 0, 0, 0, 0, 1/12, 1/12, 1/12, 1/12)</td>
<td>Station 1 M3 Station 1 M3</td>
<td></td>
</tr>
<tr>
<td>Q = (1/16, 1/16, 1/16, 1/16, 1/16, 1/16, 1/16, 1/16, 1/16, 1/16, 1/16, 1/16)</td>
<td>Station 1 M3 Station 1 M3</td>
<td></td>
</tr>
<tr>
<td>Q = (0.001, 0, 0, 0.003, 0.004, 0.004, 0, 0.014, 0.017, 0, 0.057, 0 0.112, 0, 0.323, 0.465)</td>
<td>Station 1 M3 Station 1 M3</td>
<td></td>
</tr>
</tbody>
</table>
addition, the uneven splitting of the three variants will also cause the loss of throughput because the throughput of each parallel station can not exceed 0.5 part/min. So assembling module 1 and 2 at parallel stations is not the optimal. From Table 2.3, we can see that the optimal MAS chooses to split module 1 and module 4 at parallel stations, which makes the even splitting of production at two parallel stations. When all 16 variants are offered, the optimal MAS chooses splitting the variants at parallel stations to reduce the complexity at station 3 and 4, which in turn reduces the complexity and increases the throughput of MASs. For the case of general demand, the MAS with maximum throughput is based on the combination of completely splitting of variant 1, 2, 3 and the partially splitting of variant 4 into two parallel stations. It is because the optimal MAS with maximum throughput needs to find the balance between the reduction of station level complexity from splitting the production of variants and the throughput loss caused by the uneven production splitting.

2.7 Conclusions

In this chapter, we define a complexity measure for MASs with serial, parallel and hybrid configurations. We also develop an approximate throughput model for MASs incorporating complexity based on human reaction time and fatigue effect. Then we study the MASs with different configurations by comparing complexity and throughput. We find out that the complexity increases as the configuration changes from serial to hybrid and then to parallel. We also show that variety-induced complexity influences the station reliability and in turn impacts the throughput of MASs. A MAS with higher complexity is impacted more significantly than the ones with lower complexity. This is because number of modules assembled at the stations of MASs with higher complexity is more than other MASs, which results in higher station complexity and lower reliability and in turn reduces the throughput. In addition, we discuss how to differentiate the production of variants at parallel stations to reduce the complexity of MASs and increase the throughput, because the presence of product variety in mixed-model assembly systems allows different parallel stations to produce different product variants. Based on mixed-binary nonlinear programming (MBNP), we develop the mathematical formulation to find the MAS with minimum complexity and
the MAS with maximum throughput within one configuration. This research brings new insights on how product variety induced complexity impacts the performance of MASs. The developed methodology and analysis results can be used as tools for system designers in evaluating the performance of mixed-model assembly systems with different configurations.
BIBLIOGRAPHY


CHAPTER 3

A COMPLEXITY MEASURE FOR ASSEMBLY SUPPLY CHAINS AND ITS RELATIONSHIP TO COST

ABSTRACT

We propose a complexity measure for assembly supply chains, based on the concept of information entropy. This complexity measure takes into account factors such as the supply chain configuration, the level of variety offered by each node of the supply chain, and the demand ratios across all the variants offered by a node. We investigate the relationship between the complexity and the cost of an assembly supply chain. We first study the degree of consistency between the cost and complexity criteria when comparing assembly supply chains with the same configuration but different levels of product variety. We show that the cost and the complexity are equivalent under certain conditions, in the sense that both of them rank a given set of supply chains in the same order. Even when these conditions do not hold, our numerical study demonstrates that the cost and complexity criteria rank supply chains consistently in an overwhelming majority of cases. Furthermore, the inconsistencies occur mostly when the supply chains have very small cost differences. We then study how well the cost and complexity criteria agree when comparing assembly supply chains with the same level of product variety, but different configurations. In such cases, we show that the

consistency between cost and complexity criteria is not very reliable. Overall, the results suggest that the complexity and cost criteria agree well when evaluating alternative levels of product variety that will be delivered by a given supply chain configuration, but when evaluating alternative supply chain configurations to deliver a given level of product variety, the results of complexity and cost criteria can sometimes be different.

3.1 Introduction

In many industries, previous decades brought about an explosion of product variety. For example, the number of distinct vehicle models offered in the US rose from 44 in 1969 to 168 in 2005 (Ward’s Automotive Yearbook, 1970 & 2006). The number of styles of running shoes went from five in the early 70s to 285 in the late 90s (1998 Annual Report of the Federal Reserve Bank of Dallas). The growth of product variety brings with it many challenges. For instance, several studies suggest that high product variety has a negative effect on manufacturing performance such as increasing manufacturing complexity, lowering productivity and degrading quality (MacDuffie et al., 1996; Fisher and Ittner, 1999). Many manufacturing firms have adopted modular product designs to cope with the challenges posed by high variety. A modular design decomposes the product into several modules with standard interfaces, and high product variety is achieved through the combinational assembly of different modules. In doing so, modular designs support a large variety of end products while still maintaining high volume for each module, thereby creating economies of scale in production of modules and components (see, for example, Swaminathan, 2001).

One important trend in supply chain management, enabled by modular product designs, is the emergence of modular assembly supply chains. In a modular assembly supply chain, the manufacturer apportions the product into different modules, most of which are outsourced to and assembled by suppliers. The manufacturer does only the final assembly of a few modules. For instance, Volvo’s S80 model is assembled from 23 different modules, delivered directly to the final assembly line by 17 different assembly units, 11 of which are operated by suppliers (Fredriksson, 2006).

In many instances, most notably when bringing a new product to the market, a manufac-
turing firm must make design decisions about how to modularize the product and what level of variety to offer for each module. Of course, such modularization and variety decisions influence the cost to be incurred across the entire supply chain. Thus, a manufacturing firm would benefit from getting a handle on the costs that correspond to different modularization and variety decisions. However, cost models require the estimation of many parameters, e.g., manufacturing costs, holding and shortage costs, transportation costs, production and transportation leadtimes. Furthermore, sophisticated cost models present many analytical challenges due to the network structure of an assembly supply chain, and even more so when product variety is taken into account. Hence, such cost models are difficult to develop, analyze and utilize when making modularization and variety decisions in assembly supply chains.

In this paper, we propose a new performance measure for assembly supply chains: complexity. This complexity measure is based on information entropy as applied to an assembly supply chain and recognizes the product variety to be offered by the manufacturer. We explore if, when and to what degree this complexity measure is consistent with cost. In particular, we illustrate the usefulness of this complexity measure by considering two different scenarios. First, we consider a manufacturer who has already settled on a given network of suppliers, but needs to compare different levels of product variety to be offered by the supply chain. We then analyze the agreement between the cost and complexity criteria when evaluating different levels of product variety. Second, we consider a manufacturer that has already decided to offer a given assortment of variants, but needs to choose whether to use a modular or non-modular assembly supply chain to deliver that assortment. Once again, we analyze the agreement between the cost and complexity criteria when choosing from a modular or non-modular supply chain.

When comparing supply chains with the same configuration but different levels of product variety, we show that the cost and the complexity are equivalent under certain conditions, in the sense that both of them rank a given set of supply chains in the same order. We also conduct an extensive numerical study to check how the agreement between cost and complexity is affected by different factors (e.g., number of echelons, number of suppliers, the distribution of consumer demand across variants offered by the manufacturer, etc.). Even
when the sufficient conditions for equivalence do not hold, our numerical study demonstrates that the cost and complexity criteria rank supply chains consistently in an overwhelming majority of cases. Furthermore, the inconsistencies occur mostly when the supply chains have very small cost differences. In contrast, when comparing supply chains with the same level of product variety, but different configurations, we show that the consistency between the cost and complexity criteria is not very reliable. Overall, our results suggest that the complexity measure is a good proxy for the cost of an assembly supply chain when evaluating alternative levels of product variety that will be delivered by a given supply chain configuration, but not so when evaluating alternative supply chain configurations to deliver a given level of product variety.

The paper is organized as follows. In Section 2, we review the related literature. In Section 3, we introduce our complexity measure and we describe the cost model that will serve as a benchmark. We then study the relationship between cost and complexity as a function of variety (in Section 4) and as a function of supply chain configuration (in Section 5). Section 6 concludes the paper. All the proofs are included in Appendix B.

3.2 Literature Review

An assembly system is a tree-structured network, where each node produces a single item and has at most one successor. Early research on assembly systems seeks to characterize the optimal ordering policy at each node of the system to minimize the total system cost. Schmidt and Nahmias (1985) use a finite-horizon model to explore an assembly system with two components. Rosling (1989) studies the periodic review, infinite-horizon inventory problem and shows that the assembly system can be reduced to an equivalent serial system. Subsequent research has explored more elaborate assembly systems, for example, assembly systems in which not only the demand but also the supply may be random (e.g. Gurnani et al., 1996; Bollapragada et al., 2004), assembly systems in which inventory levels may be supplemented with returns (e.g. DeCroix and Zipkin, 2005) and decentralized assembly systems (e.g. Bernstein and DeCroix, 2006; Gerchak and Wang, 2004). In contrast to an assembly system, an assemble-to-order system is traditionally defined as a two-echelon
system, in which multiple products are assembled from a given set of components. For a review of the research on assemble-to-order systems, see Song and Zipkin (2003). Assemble-to-order systems present one additional challenge that is not present in the single-product assembly system: allocating a component’s inventory among several different products that use the component (hence, the need to restrict attention to two-echelon systems).

The assembly supply chain we consider in this paper has elements of both an assembly system (in that the supply chain has the same tree structure as the assembly system) and an assemble-to-order system (in that each node of the supply chain produces multiple variants of its output, each of which goes into multiple items in the downstream node). In a supply chain as complicated as the one considered in this paper, even when one limits attention to a given ordering policy, it is still very challenging to obtain an exact expression for inventory costs. As our main focus in this paper is to introduce a complexity measure, we do not attempt to develop an exact expression for the supply chain costs. To analyze the agreement between cost and complexity, we make simplifying assumptions to obtain only a rough estimate of the costs that will be incurred in the supply chain. In particular, we assume the nodes in the assembly supply chain can be decoupled, after which each node follows an order-up-to policy to satisfy a certain service level objective.

One application of the proposed complexity measure is to make product variety decisions for an assembly supply chain with a given network of suppliers. The operations literature has given scant attention to the management of product variety in a supply chain. Kurtulus and Toktay (2005) analyze if and when a retailer should delegate variety and pricing decisions to a manufacturer. Singh et al. (2005) examine the effect of supply chain structure, in particular, the effect of drop-shipping, on the optimal assortment. Aydin and Hausman (2009) consider the coordination of product variety decisions in a single-retailer, single-manufacturer supply chain. Similar to these papers, we use a demand model where the customer demand is allocated among many variants of a product. Given that we are modeling an assembly supply chain, the supply relationships in our model are significantly more complicated than in this earlier work. Instead of developing a detailed analysis of cost-minimizing or profit-maximizing variety levels, we focus on the use of complexity measure to evaluate alternative levels of product variety. In that sense, our work complements
the earlier literature by showing the usefulness of complexity measure when choosing the level of variety.

Another application of the proposed complexity measure is to compare the efficiency of two different supply chain structures: modular versus non-modular assembly supply chains. Such supply chain configuration decisions received limited attention in the literature. Bernstein and DeCroix (2004) analyze a modular assembly supply chain to determine the optimal transfer prices between the manufacturers and subassemblers. They identify if and when a modular assembly supply chain is more beneficial than a non-modular assembly supply chain. Thomas and Warsing (2007) consider a service supply chain in which there is external demand not only for the modular end product, but also for the components that make up the end product. They evaluate the savings that could be obtained if the supply chain adjusted its inventory levels not only by placing orders with external vendors, but also by utilizing assembly and disassembly operations to shift the inventory from the end product to the components or vice versa. Taking an empirical approach, Randall and Ulrich (2001) use data from U.S. bicycle industry to examine the relationship among product variety, supply chain structure and system performance. They show that firms that match their supply chain structure to the type of variety they offer often outperform those that fail to make such choices, where the performance is measured based on cost and revenue analysis. Salvador et al. (2004) also use 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. In a similar vein, we compare modular and non-modular assembly supply chains, with the goal of illustrating the usefulness of the complexity measure.

With different applications in mind, several different definitions of complexity have been proposed by researchers. See, for example, Suh (2005) for a complexity definition particularly applicable in product design, and Cover and Thomas (1991) for a discussion of 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 (1948) in the context of communication systems. Shannon’s information entropy is a measure of the uncertainty surrounding the outcome of a ran-
dom experiment. The information entropy has been used to study complexity in several different areas, including communication networks, biology, and physics. Shannon’s information entropy has been used to measure the complexity of manufacturing systems as well. Deshmukh et al. (1998) derive 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. (2008) study the operator choice complexity in mixed model assembly lines and develop a methodology to find the optimal assembly sequence to minimize manufacturing complexity.

### 3.3 Complexity Model of An Assembly Supply Chain

Consider an assembly supply chain, where each node can have multiple suppliers, but a given node cannot be a supplier to multiple nodes. Suppose there are \( n \) nodes in the assembly supply chain. As a convention, we let node \( n \) be the final assembler. In keeping with our focus on product variety, we assume that each node in the most upstream echelon can produce a number of variants. A downstream node can potentially assemble any combination of the variants provided by its suppliers, and each combination counts as a distinct variant. See Figure 3.1 for an illustration. We assume that if variant \( v \) of node \( j \) is used when producing variant \( u \) of node \( i \), then one unit of variant \( v \) goes into one unit of variant \( u \). This assumption is merely to simplify the exposition.

We define the following notation:

- \( V_i \): the number of variants that node \( i \) can produce, \( i = 1, \ldots, n \).
- \( S_i \): the set of nodes that are suppliers to node \( i \), \( i = 1, \ldots, n \).
- \( A_{ijv} \): the set of variants produced at node \( i \) that use variant \( v \) from node \( j \), where node \( j \) is a supplier to node \( i \), i.e., \( j \in S_i \).

Let \( q_{iv} \) denote the fraction of node-\( i \) demand that belongs to variant \( v = 1, \ldots, V_i \). Hereafter, we refer to \( q_{iv} \) as the demand share of variant \( v \) at node \( i \). In addition, define the vector \( \mathbf{q}_i := (q_{i1}, q_{i2}, \ldots, q_{iV_i}) \), which captures the mix ratio of the variants produced by node \( i \). Hereafter, we refer to \( \mathbf{q}_i \) as the demand vector of node \( i \). The final assembler’s demand vector, \( \mathbf{q}_n \), determines how the demand at all other upstream nodes is allocated.
Figure 3.1: Relationship between the demand vectors of upstream and downstream nodes. In this figure, nodes \(j\) and \(j+1\) are suppliers to node \(i\). Each of node \(j\) and \(j+1\) produces two variants, which results in node \(i\) producing four different variants of its product, where each variant corresponds to a distinct combination of the variants supplied by nodes \(j\) and \(j+1\).

among several variants produced at those nodes (see Figure 3.1). In particular, using the notation introduced so far, we have the following relationship between the demand share of variant \(v\) at node \(j\), \(q_{jv}\), and the demand vector of node \(i\), \(q_i\), where node \(j\) is a supplier to node \(i\), (i.e., \(j \in S_i\)):

\[
q_{jv} = \sum_{u \in A_{ijv}} q_{iu}, j \in S_i. \quad (3.1)
\]

A measure of the assembly supply chain performance should take into account factors such as the supply chain’s configuration, the number of variants produced at each node of the supply chain, and the demand vector of each node. In an effort to capture these factors, we use a complexity measure based on Shannon’s information entropy. The information entropy of a random experiment is a measure of the uncertainty about the outcome of the random experiment (Shannon, 1948). According to Shannon’s definition, the information entropy of a random experiment with \(r\) possible outcomes, whose probabilities of occurrence are \(p_1, p_2, ..., p_r\), is

\[
H = -\sum_{i=1}^{r} p_i \log_2 p_i, \quad (3.2)
\]

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

In what follows, we offer two different complexity measures for an assembly supply chain, node-based complexity and arc-based complexity, and we relate these complexity measures to Shannon’s information entropy.
Node-based Complexity: Let $H_N$ denote the node-based complexity of the supply chain and define it as

$$H_N = -\sum_{i=1}^{n} \sum_{v=1}^{V_i} q_{iv} \frac{q_{iv}}{n} \log_2 \frac{q_{iv}}{n}.$$  \hspace{1cm} (3.3)

One possible interpretation of this complexity measure is the following. Suppose, for each node, we form a pool of variants produced by that node, where each variant is represented in a quantity proportional to its demand share. Consider the random experiment where we first pick a node at random, and we then pick one item from this node’s pool. The probability of picking variant $v$ of node $i$ is $q_{iv}/n$ (since we pick node $i$ with probability $1/n$ and variant $v$ of node $i$ with probability $q_{iv}$). The information entropy of this random experiment is given by (3.3) and yields our node-based complexity measure. Loosely speaking, the node-based complexity indicates the level of uncertainty about what variant in the supply chain will be demanded next.

Arc-based Complexity: The node-based complexity has the attractive property that it takes into account the number of nodes in the supply chain as well as the split of demand among the variants. However, there is one aspect of the supply chain that is ignored by the previous definition. Two supply chains may have the same number of nodes, but they can be very different from one another due to the configuration of supply relationships among the nodes. See, for example, Figure 3.2 for an illustration. In order to better capture the information about supply chain configuration, we offer an alternative complexity measure, the arc-based complexity. Let $L_i$ denote the number of suppliers of node $i$, i.e., $L_i \equiv |S_i|$. In order to have a complete picture of the flows into the supply chain, we also assume that there is a virtual supplier that is linked to each of the nodes in the most upstream echelon of the supply chain. Hence, $L_i = 1$ for all suppliers $i$ in the most upstream echelon. Let $K$ denote the number of arcs in this supply chain, including those that connect the virtual supplier with the suppliers in the most upstream echelon, i.e., $K = \sum_{i=1}^{n} L_i$. We let $H_A$ denote the arc-based complexity and define it as follows:

$$H_A = -\sum_{i=1}^{n} \sum_{j \in S_i} \sum_{v=1}^{V_j} q_{iv} \frac{q_{iv}}{K} \log_2 \frac{q_{iv}}{K}.$$  \hspace{1cm} (3.4)
This complexity measure can also be interpreted in the context of Shannon’s information entropy. Suppose again, for each node, we form a pool of variants produced by that node, 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 \( a \), 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 arc \( a \) with probability \( 1/K \) and variant \( v \) of node \( i \) with probability \( q_{iv} \)). It can be shown that the information entropy of this random experiment is given by (3.4), which yields our arc-based complexity measure and the arc-based complexity defined by (3.4) can be simplified as follows (see Appendix A for a derivation):

\[
H_A = -\sum_{i=1}^{n} \sum_{v=1}^{V_i} L_i \frac{q_{iv}}{K} \log_2 \frac{q_{iv}}{K}.
\] (3.5)

Loosely speaking, the arc-based complexity indicates the level of uncertainty about the next flow of material that will occur in the supply chain.

The complexity measures we propose are useful to the extent that they agree with a more direct performance measure like cost. In the next subsection, we describe a simple cost model for an assembly supply chain that will be used as a benchmark to study the effectiveness of node-based and arc-based complexity measures.

### 3.4 Cost Model of An Assembly Supply Chain

Several types of costs can be incurred in an assembly supply chain, including inventory costs, manufacturing costs and transportation costs. In this section, we describe a simple
cost model for an assembly supply chain based on inventory costs only. This simple cost model will be used to analyze the effectiveness of the complexity measure in evaluating supply chain performance. That is, we will analyze whether more costly supply chains rank higher on the complexity scale as well. In the conclusion section, we discuss certain assumptions under which our results extend to include other types of costs such as manufacturing and transportation costs.

3.4.1 Overview of the Cost Model

We assume that each node in the supply chain keeps inventory of its own variants, but no inventory of the inputs from the suppliers. Hence, every time node $i$ decides to replenish the inventory of one of its variants, it must first buy the necessary inputs and then assemble those. The assumption that the nodes do not hold any component inventory comes close to reality in environments where the supply chain partners, utilizing lean production principles, are located close to one another. At the other extreme, one could assume that all the nodes keep inventory of components, but no inventory of the finished goods. This assumption would be close to reality in environments where all supply chain partners assemble to order. All of the results in the next section continue to hold if we assume that only component inventories are held.

In an assembly system, the ordering decisions for an item tend to depend on the inventory levels for other items that it will be assembled with. This feature of an assembly system makes it challenging to characterize the optimal inventory decisions. In addition to this difficulty that is inherent in any assembly system, our assembly supply chain presents further challenges due to the presence of product variety. In our assembly supply chain, each item supplied by a node is an input to multiple variants at the downstream node. This one-to-many relationship brings up an important challenge: After receiving the delivery of an input, how should a node allocate the delivered quantity among many variants that must share the supply of this input? The allocation policy and inventory levels at upstream locations interact in complicated ways to determine the inventory costs, making it hard to analytically characterize the inventory cost of the supply chain.

In order to obtain a simple enough cost benchmark, we utilize a cost model that relies
on two simplifying assumptions. First, we assume that each node uses an order-up-to policy for each variant so as to achieve a certain a service level objective for that variant. Such policies are practical and they are commonly used. Second, and perhaps more restrictively, we assume that the appropriate order-up-to levels can be determined by decoupling the assembly supply chain into a collection of standalone nodes. We utilize the decoupling assumption to create a tractable cost benchmark. The higher the service levels across the supply chain, the closer to reality the decoupling assumption is and the better the approximations obtained. Similar assumptions have been used by others when analyzing inventory costs in the presence of product variety. For example, to model a production environment with delayed differentiation, Lee and Tang (1997) consider a model where a series of common operations eventually fork into two distinct series of operations to produce two distinct products. In their model there exists buffer inventory between each pair of operations, and the buffer at each operation needs to satisfy a certain service level objective. Lee and Tang (1997) also make a decoupling assumption, essentially assuming that the buffer size at each operation can be determined independent of other operations. We next describe the details of our cost model.

3.4.2 Derivation of the Supply Chain Cost

We assume that each node uses a periodic-review, order-up-to policy in an infinite-horizon setting with leadtime. The order-up-to level for each variant at each node is chosen to satisfy what is commonly called a type-1 service level objective, that is, the probability of meeting the demand in full in a given period. For simplicity, we assume that all variants at all nodes have the same type-1 service level objective, denoted by $\alpha$, where $0 < \alpha < 1$. It would not be difficult to extend the model so that the service level objectives differ across variants and/or nodes. The leadtime for replenishing the inventory of a variant covers the span of activities starting with the purchase of the inputs and ending with the assembly of the variant. Let $l_i$ denote the leadtime to replenish the inventory of a variant at node $i$.

The timing of events is as follows: (1) At the beginning of a period, each node $i = 1, \ldots, n$ receives the units that it ordered $l_i$ periods ago. (2) Each node then reviews the inventory positions of its variants (inventory on hand plus pipeline inventory that has been ordered
but not yet received) and orders enough of each variant to bring its inventory position to the desired order-up-to level. (3) The demand for each variant at each node is realized.

(4) After the demands are realized, node $i$ incurs an overage cost of $c_{io}$ per unit of leftover inventory and a unit underage cost of $c_{iu}$ per unit of shortage. We assume that all the unmet demand is backordered. Notice our implicit assumption that the underage and overage costs are the same across all the variants of node $i$.

As for the demand model, suppose that the total per-period demand faced by the final assembler (node $n$) follows a Poisson distribution with rate $\lambda$. Given the final assembler’s demand vector, denoted by $q_n := (q_{n1}, q_{n2}, \cdots, q_{nV_n})$, the demand for variant $v$ of the final assembler also follows a Poisson distribution with rate $\lambda q_{nv}$, $v = 1, 2, \cdots, V_n$. Furthermore, the per-period demands are independent across variants. Hence, using the normal approximation to Poisson random variables, we assume that the per-period demand for variant $v$ of the final assembler is normal with mean and variance $\lambda q_{nv}$, $v = 1, \ldots, V_n$, and the per-period demands are independent across variants. This approximation is valid when $\lambda$ is large. Furthermore, for analytical convenience, we assume that the per-period demands of a variant are independent and identically distributed (i.i.d.) over periods. Similarly, for the remaining nodes 1 through $n-1$, we assume that the per-period demand for variant $v$ of node $j$ is normally distributed with a mean and variance $\lambda q_{jv}$, $v = 1, \ldots, V_j$, where $q_{jv}$ is given by (3.1). Notice that the final assembler’s demand vector influences both the mean and variance of the demands at upper echelons, since the demand shares of variants at upper echelons are determined by the final assembler’s demand vector. We assume that, at each of nodes 1 through $n-1$, the demands are independent across variants and, for a given variant, the demands are independent across periods.

Using standard arguments from inventory theory, the order-up-to level for variant $v$ of node $i$ is given by

$$y_{iv}^* = (l_i + 1)\lambda q_{iv} + z(\alpha)\sqrt{(l_i + 1)\lambda q_{iv}},$$

(3.6)

where $z(\alpha)$ is $\alpha$-fractile of the standard normal distribution, i.e., $z(\alpha)$ is such that $\Phi_N(z(\alpha)) = \alpha$ where $\Phi_N(\cdot)$ is the cumulative distribution function (c.d.f.) of the standard normal distribution. Furthermore, the expected per-period inventory cost for variant $v$ of node $i$ is
given by

$$I_{iv} = \sqrt{(l_i + 1)\lambda q_{iv}} \left[ (c_{io} + c_{iu})\alpha z(\alpha) - c_{iu}z(\alpha) + (c_{io} + c_{iu})\phi_N(z(\alpha)) \right]$$  \hspace{1cm} (3.7)$$

where $\phi_N(\cdot)$ is the probability density function (p.d.f.) of the standard normal distribution. For notational convenience, we let

$$C_i := (c_{io} + c_{iu})\alpha z(\alpha) - c_{iu}z(\alpha) + (c_{io} + c_{iu})\phi_N(z(\alpha)),$$

and we refer to $C_i$ as the cost coefficient of node $i$. Then, the expected per-period inventory cost of the entire assembly supply chain is given by

$$I = \sum_{i=1}^{n} \sum_{v=1}^{V_i} I_{iv} = \sum_{i=1}^{n} \sum_{v=1}^{V_i} C_i \sqrt{(l_i + 1)\lambda q_{iv}}.$$  \hspace{1cm} (3.8)$$

Thanks, in particular, to our decoupling assumption, this cost expression is simple. The simplicity of the cost expression, however, does not necessarily make it an easy tool for practical purposes. To obtain the supply chain costs through this expression, one would need to estimate the unit underage cost, the unit overage cost and the leadtime for each node of the supply chain. In contrast, the complexity measures we introduced earlier do not require the estimation of these parameters.

### 3.5 Complexity versus Cost When Choosing Variety

Consider a final assembler who is committed to working with a given network of suppliers and needs to decide what level of variety to offer through this already finalized supply chain. In this section, we analyze if, when and to what degree the cost and the complexity criteria lead to consistent recommendations for such a supply chain. Different variety decisions lead to different demand vectors. Hence, we ask the following question: Given a set of supply chains that share the same configuration but that differ from one another in the number of variants and demand vectors, does the complexity criterion rank these supply chains in the same order as the cost criterion would? First, we show that, under certain
conditions, complexity and cost are equivalent. We then conduct an extensive numerical
study to further investigate the agreement between the cost and complexity.

3.5.1 Conditions for Equivalence between Complexity and Cost

To help compare supply chains, we define some additional notation. Suppose we have
m supply chains to compare. Let $V_i^k$ be the number of variants offered by node $i = 1, \ldots, n$
in supply chain $k = 1, \ldots, m$. Let $q_i^k$ denote the demand vector at node $i$ of supply chain
$k$. In addition, let $I^k$, $H_N^k$ and $H_A^k$ denote, respectively, the cost, node-based complexity
and arc-based complexity of supply chain $k$. The following proposition states one condition
under which the cost and complexity are equivalent.

**Proposition 3.1** Consider a set of $m$ supply chains that have the same configuration but
differ from one another in the number of variants and the demand vectors of the nodes.
Suppose all $m$ supply chains have the property that the demand shares are equal across all
variants of the final assembler, i.e., $q_{nj}^k = \frac{1}{V_n^k}$ for $j = 1, \ldots, V_n^k$. The ordering of these
supply chains according to cost is the same as the ordering of them according to node-based
complexity and arc-based complexity, i.e., $I^k > I^l$ iff $H_N^k > H_N^l$ and $I^k > I^l$ iff $H_A^k > H_A^l$.

Proposition 3.1 implies that, under the condition that demand shares at the final assembler
are evenly distributed, both the cost criterion and the complexity criterion will rank a given
set of supply chains in the same order. In what follows, we show such equivalence holds
under a less restrictive condition as well. Consider the case where the demand shares of all
the variants offered by the final assembler are the same except for one dominant variant
whose demand share is larger than all the others. In this case, the equivalence between the
cost and the complexity continues to hold.

**Proposition 3.2** Consider a set of $m$ supply chains that use the same configuration and
offer the same set of variants, but differ from one another in the demand vectors of their
nodes. Suppose all supply chains have the property that the demand shares of all variants
of the final assembler are equal except for one dominant variant (indexed to be variant
1), whose demand share is larger than the demand shares of all the other variants, i.e.,
$q_{n1}^k \geq q_{n2}^k = \ldots = q_{n,V_n^k}^k$ for $k = 1, \ldots, m$. The ordering of these supply chains according to
cost is the same as the ordering of them according to node-based complexity and arc-based complexity, i.e., \( I^k > I^l \) iff \( H^k_N > H^l_N \) and \( I^k > I^l \) iff \( H^k_A > H^l_A \).

To obtain another generalization on when the cost and the complexity are equivalent, we turn to majorization theory, which is useful in comparing how disordered two vectors are.

In preparation for our next result, we first provide a formal definition of majorization.

Let \( x = (x_1, x_2, ..., x_d) \) and \( y = (y_1, y_2, ..., y_d) \) be two \( d \) dimensional real vectors. Let \((x(1), x(2), ..., x(d))\) indicate the vector obtained by sorting the entries of vector \( x \) in decreasing order. We say \( y \) majorizes \( x \), denoted as \( y \succ x \), if the following two conditions are satisfied:

\[
\sum_{i=1}^{k} x(i) \leq \sum_{i=1}^{k} y(i), k = 1, 2, ..., d - 1, \quad \text{and} \quad \sum_{i=1}^{d} x(i) = \sum_{i=1}^{d} y(i). \tag{3.9}
\]

In our setting, majorization has a meaningful interpretation: If one demand vector majorizes another, the majorizing vector represents a more predictable demand pattern than the majorized vector. This interpretation has been used in the literature when analyzing the effect of variety on inventory costs; for example, van Ryzin and Mahajan (1999) interpret that a majorized demand vector represents a more fashionable product’s demand, where consumer choice is less predictable and, therefore, more evenly scattered across variants.

In our context, majorization provides a meaningful way to compare supply chains through their demand vectors. The next proposition utilizes the majorization theory to describe another scenario where the cost and complexity criteria are equivalent.

**Proposition 3.3** Consider a set of \( m \) supply chains that have the same configuration but differ from one another in the demand vectors of the final assemblers. Suppose the supply chains have the property that, for any given node in the uppermost echelon, the demand vector of supply chain 1 majorizes that of supply chain 2 which majorizes that of supply chain 3 and so on, i.e., \( q^1_i \succ q^2_i \succ \ldots \succ q^m_i \) for all nodes \( i \) in the uppermost echelon.

The ordering of these supply chains according to cost is the same as the ordering of them according to node-based complexity and arc-based complexity, i.e., \( I^k > I^l \) iff \( H^k_N > H^l_N \) and \( I^k > I^l \) iff \( H^k_A > H^l_A \).

One interpretation of Proposition 3.3 is the following: If a set of supply chains can be ordered according to the demand variability faced by the most-upstream suppliers, then the cost
and complexity criteria will rank these supply chains in the same order. The underpinnings of this result are two intuitive observations: First, loosely speaking, the higher the demand variability faced by a supplier, the higher the cost incurred by the supplier and the higher the contribution of this supplier to the complexity of the supply chain. In that sense, higher cost and higher complexity go hand in hand as demand variability increases. Second, as we show in the proof of Proposition 3.3, if the uppermost echelons of supply chains can be ordered according to their demand variability, the same ordering cascades down the supply chain to lower echelons, which essentially means that the supply chains themselves can be ordered according to the demand variability they face. As a consequence, supply chains with higher demand variability end up being simultaneously more costly and more complex.

Given Proposition 3.3, one may wonder if a similar result holds when the demand vectors of the final assemblers are ordered according to the majorization criterion. In other words, is the following claim true?

**Claim:** Assume supply chains A and B have the same configuration. If the final assembler’s demand vector in supply chain A majorizes the final assembler’s demand vector in supply chain B, then the supply chain with the higher cost also has higher complexity.

Unfortunately, the above claim is not true. The condition that the demand vector of one final assembler majorizes that of another is not strong enough, because such an ordering does not propagate up the supply chain to upper echelons, making it impossible to say that one supply chain faces more demand variability than the other. Figure 3.3 shows a counterexample to the above claim. In this example, the demand vector of the final assembler in supply chain A majorizes the demand vector of the final assembler in supply chain B. Here, supply chain A has lower cost, but higher complexity.

The counterexample shows that there are instances where the supply chain with higher complexity could have lower cost, causing an inconsistency between how the cost and the complexity criteria rank supply chains. This is not surprising given that the cost and the complexity are two entirely different functions. Nonetheless, the results of this section show that the cost and complexity share a similar enough structure that they will rank a given set of supply chains consistently under certain conditions. Next, we conduct a numerical study to further investigate the the degree of agreement between cost and complexity.
Figure 3.3: An example where complexity and cost are inconsistent. Here $q^A_3 \succ q^B_3$, supply chain B has higher cost but with lower complexity. Both supply chain A and B have the same cost coefficients $C_1 = C_2 = 1$ and $C_3 = 8$

3.5.2 Numerical Analysis of the Consistency between Complexity and Cost

We first investigate how five different factors affect the consistency between the complexity and the cost of a two-echelon supply chain. These five factors are: complexity definition, number of suppliers to the final assembler, underage and overage cost difference between two consecutive echelons, number of variants offered by suppliers in the uppermost echelon, and the evenness of the final assembler’s demand vector. We first present our results for a two-echelon supply chain and subsequently analyze the effect of the number of echelons.

Consider two supply chains that differ from one another only in their demand vectors. (One can generate many such supply chain pairs by randomly generating a demand vector for each of the two final assemblers, which then determines the demand vectors of all the other nodes.) For a given pair of supply chains, if the supply chain with higher cost has higher complexity as well, we say cost and complexity are consistent. Otherwise, we say cost and complexity are inconsistent. By checking many pairs of supply chains, one can observe how common inconsistencies are and how the likelihood of an inconsistency depends on the five factors identified above. To perform a systematic analysis, we conduct a numerical experiment using a $2^5$ full factorial design (where each of the five factors can take two values and all 32 combinations of factor values are considered). Table 3.1 shows the levels allowed for each factor in our experiment, which we discuss next:

---

2 A full two-level factorial design is an experiment in which several factors, each of which can take two values, vary independently, and the experiment is conducted for all possible combinations of the levels across all such factors. It is used to study the effect of each factor on the response variable, as well as the effects of interactions between factors on the response variable (Wu and Hamada, 2000).
First, the complexity definition can be one of two types: Node-based or arc-based complexity.

Second, the number of suppliers to the final assembler is either two or three.

Third, we assume that all nodes in the uppermost echelon provide the same number of variants, which can be either two or three.

Fourth, we let the underage and overage cost difference between the two echelons to be high or low. We refer to this factor as cost disparity hereafter. To obtain high and low levels for cost disparity, we start by assuming that the final assembler’s cost coefficient is at least as big as the sum of the suppliers’ cost coefficients. Such an assumption is justified since the cost coefficient is based on underage and overage costs, and the final assembler puts together components from suppliers and adds further value to the product, thus inflating the overage and underage costs beyond those of the components from the suppliers. We say the cost disparity is low if the final assembler’s cost coefficient is the sum of the suppliers’ cost coefficients, and high if the final assembler’s cost coefficient is twice the sum of the suppliers’ cost coefficients.

The fifth and last factor in our experiment is the evenness of the final assembler’s demand vector. To obtain two different degrees of evenness, we use the following procedure to generate the demand vectors. Given that the final assembler offers \( V_n \) variants, we draw \( V_n \) random numbers, denoted as \( R_v, v = 1, 2, ..., V_n \), from the uniform distribution, \( U(a-Kb, a+Kb) \) (where \( a > Kb > 0 \) so that all the random numbers are positive). We then obtain the final assembler’s demand vector \( q_n \) from random numbers \( R_v, v = 1, 2, ..., V_n \) by letting \( q_{nv} := \frac{R_v}{\sum_{k=1}^{V_n} R_v} \). Given this procedure, observe that the choice of \( K \) influences how even the resulting demand vector will be. We let \( K = 1 \) to obtain a more even demand vector and \( K = 8 \) to obtain a less even demand vector.

Under each combination of the five factors, 10,000 pairs of supply chains are generated where each pair consists of two supply chains that differ from one another in terms of their demand vectors. For each supply chain pair, we determine whether cost and complexity are consistent. We then determine the inconsistency percentage among these 10,000 pairs. This entire experiment (10,000 pairs for each of 32 combinations) is then replicated three times.
Table 3.1: Design of Experiments. We use a two-level, full factorial experiment.

<table>
<thead>
<tr>
<th>Factor</th>
<th>Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity definition</td>
<td>Arc-based Complexity</td>
</tr>
<tr>
<td></td>
<td>Node-based Complexity</td>
</tr>
<tr>
<td>1. Complexity definition</td>
<td>-</td>
</tr>
<tr>
<td>2. Number of suppliers</td>
<td>+</td>
</tr>
<tr>
<td>3. Number of variants</td>
<td>2</td>
</tr>
<tr>
<td>4. Cost disparity between two echelons</td>
<td>3</td>
</tr>
<tr>
<td>5. Evenness of the demand vector</td>
<td>C_i = \sum_{j \in S_i} C_j</td>
</tr>
<tr>
<td></td>
<td>K=8</td>
</tr>
<tr>
<td></td>
<td>C_i = 2 \sum_{j \in S_i} C_j</td>
</tr>
<tr>
<td></td>
<td>K=1</td>
</tr>
</tbody>
</table>

In this numerical study we fix the leadtime parameters so that the leadtime to replenish the inventory of a variant at node \(i\) is \(L_i\) periods, where \(L_i\) is the number of suppliers to node \(i\) (equivalently, the number of inputs that are assembled by node \(i\)). Recall that the leadtime covers the span of activities starting with the purchase of the inputs and ending with the assembly of the variant. Therefore, one would expect that the larger the number of inputs that go into a variant, the longer the time it takes to replenish the variant’s inventory. This effect is what we wish to capture in a simple way through our assumption that the replenishment leadtime for a variant is equal to the number of inputs that go into the variant.

Table 3.5 in Appendix C shows the inconsistency percentage, also called inconsistency rate, for each replication of each of the 32 combinations. Using this data, we determine the statistical significance of each of the five factors (at a 99% confidence level), reported in Table 3.2. The statistical analysis is based on ANOVA (analysis of variance). For a given factor, the effect column in Table 3.2 is the change in the average inconsistency rate when the factor’s value changes from ‘-’ to ‘+’. For example, the average inconsistency rate increases by 0.00635% when the number of suppliers increases from two to three. We next discuss the results for each of the five factors listed earlier.

First, the complexity definition has a statistically significant effect on the inconsistency rate. Observe from Table 3.2 that the inconsistency rate is lower under arc-based complexity than under node-based complexity. In that sense, the arc-based complexity is a better performance measure of an assembly supply chain than node-based complexity. To see why this is the case, first recall our assumption that a node with a larger number suppliers faces a larger lead time, resulting in higher costs for those nodes. Hence, nodes with more suppliers
Table 3.2: Estimated effects, t-Statistics and p-Value (%). A positive (negative) effect implies that when the factor’s value changes from ‘-’ to ‘+’, the inconsistency rate increases (decreases).

<table>
<thead>
<tr>
<th>Term</th>
<th>Effect</th>
<th>t-Statistics</th>
<th>p Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>130.94</td>
<td>&lt; 0.0001</td>
<td></td>
</tr>
<tr>
<td>Complexity definition</td>
<td>0.00659</td>
<td>18.89</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>Number of suppliers</td>
<td>0.00635</td>
<td>18.20</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>Number of variants</td>
<td>0.00076</td>
<td>2.18</td>
<td>0.033</td>
</tr>
<tr>
<td>Cost disparity between two echelons</td>
<td>0.00252</td>
<td>7.22</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>Uniformity of the demand vector</td>
<td>-0.02700</td>
<td>-77.35</td>
<td>&lt; 0.0001</td>
</tr>
</tbody>
</table>

contribute more to the supply chain cost. Now, notice that, under node-based complexity, given by (3.3), the contributions of all nodes to the supply chain complexity are weighted equally, whereas, under arc-based complexity, given by (3.5), node \( i \)’s contribution to the supply chain complexity is weighted by the number of suppliers to that node, \( L_i \). Because arc-based complexity gives more weight to nodes with larger number of suppliers, it comes closer to capturing the supply chain cost compared to node-based complexity.

Second, the number of suppliers is also statistically significant, and an increase in the number of suppliers results in an increase in the inconsistency rate. As the number of suppliers further increases, one would hope that the inconsistency rate does not grow indefinitely, but stabilizes instead. To resolve this question, we re-ran our experiment with two new values for the number of suppliers: four and five (as opposed to two and three suppliers in the original experiment). For this experiment where the number of suppliers can be either four or five, the inconsistency data and the results of the statistical analysis are summarized in Tables 3.6 and 3.7 in Appendix C. When the number of suppliers goes from four to five, the number of suppliers no longer has a statistically significant effect on the inconsistency rate. This is encouraging because it implies that there is a natural bound on how large the inconsistency rate can grow as the number of suppliers increases.

Third, the effect of the number of variants is not statistically significant.

Fourth, the cost disparity between two echelons is statistically significant and the higher the cost disparity, the higher the inconsistency rate. However, similar to the effect of the number of suppliers, the effect of the cost disparity also becomes statistically insignificant once the cost disparity becomes large enough. For an experiment where the cost coefficient
of the final assembler is either four or five times as large as the sum of the suppliers’
cost coefficients (as opposed to the original experiment where the final assembler’s cost
coefficient was either equal to the sum of the suppliers’ cost coefficients or twice as large),
the inconsistency data and the results of the statistical analysis are summarized in Tables
3.8 and 3.9 in Appendix C.

Finally, the evenness of the final assembler’s demand vector has a statistically significant
effect on the inconsistency rate: the more even the demand vector, the smaller the inconsis-
tency rate. To see the intuition behind this, recall from Proposition 3.1 that, if all variants
of the final assembler have the same demand share, then cost and complexity are equivalent.
The larger the evenness of the final assembler’s demand vector, the closer one comes to the
scenario of evenly-distributed demand, where cost and complexity are equivalent.

We have so far focused on a numerical experiment where supply chains have two echelons.
To determine the effect of the number of echelons, we run a similar experiment with five
factors. In this new experiment, we fix the number of suppliers for each node as two (thus,
dropping the number of suppliers from the list of factors). Instead, we add the number of
echelons as a new factor and we allow the number of echelons to be two or three, shown
in Figure 3.4. Tables 3.10 and 3.11 in Appendix C show the inconsistency data of this
experiment and the results of the significance test. Observe that the number of echelons
increasing from two three has a statistically significant effect on the inconsistency rate.
Thus, one should be more careful about applying the complexity measure to supply chains
with larger number of echelons. However, it should be noted that the magnitude of increase
in inconsistency rate is is so small that the number of echelons does not appear to be a major
cause for concern: The inconsistency percentage increases by 0.198% when the number of
echelons increases from two to three.

The Cost of Inconsistencies

The previous discussion shows that inconsistencies between cost and complexity occur
rarely. Even though inconsistencies are rare, complexity may still be unreliable if it favors
a supply chain that is much more costly than another. We next analyze all the cases where
cost and complexity were inconsistent (among three replications of 10,000 examples for each
of 32 different factor value combinations, resulting in a total of 960,000 examples). For each
example where cost and complexity were inconsistent, we check the cost difference between the two supply chains. The mean, median, minimum, maximum and standard deviation of these cost differences (expressed as a percentage of the less costly supply chain’s cost) are shown in Table 3.3. Notice that the largest cost difference ever encountered is less than 1.21%. This provides good support for complexity as a measure of the supply chain performance, because what we see here is that when cost and complexity are inconsistent, the costs of the two supply chains are virtually the same. In addition, Figure 3.5 shows a frequency diagram for the cost differences (again, expressed as a percentage). Notice that the frequency diagram exhibits a long tail ending at the maximum cost difference of 1.21% and more than half of the inconsistencies occur when cost difference is less than 0.1%.

Table 3.3: Cost difference statistics (%).

<table>
<thead>
<tr>
<th>median</th>
<th>mean</th>
<th>standard deviation</th>
<th>max</th>
<th>min</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0583</td>
<td>0.1012</td>
<td>0.1286</td>
<td>1.2089</td>
<td>4.68 × 10⁻⁸</td>
</tr>
</tbody>
</table>

It is remarkable that cost and complexity disagree so rarely (2.29% of all examples), and when they do disagree, the cost difference is tiny as discussed above. One may wonder if such strong consistency is an artefact of our numerical set-up. In particular, in our experiments, when randomly generating the demand vector for a supply chain, we draw random numbers from a uniform distribution \( U(a - Kb, a + Kb) \). Because all the random numbers are coming from the same distribution, our numerical set-up might be introducing a bias toward more even demand vectors, which would in turn reduce the extent of inconsistencies. To remove this possible bias, we modify our numerical experiment by introducing deliberately large differences between the random numbers that are used to generate the demand vectors. To do so, when generating a set of \( V_n \) random numbers, denoted as \( R_v, v = 1, 2, \ldots, V_n \), we
Figure 3.5: Histogram of cost difference. Considering all the problem instances where cost
and complexity are inconsistent, more than 80% of those instances occur when the cost
difference is less than 0.2%.

draw $R_v$ from a uniform distribution $U(a_v - Kb, a_v + Kb)$, where $a_v > Kb > 0$ and $a_v = e^v,
v = 1, 2, \cdots, V_n$. Notice that the expected value of $R_v$ is now $a_v$, which grows exponentially
in $v$. This numerical set-up ensures that there will be big differences between $R_v$ values,
which will then lead to uneven demand vectors. Given this method of generating random
demand vectors, we run another experiment with a $2^4$ full factorial design, where the four
factors are: complexity definition, number of suppliers, number of variants, cost disparity.
The values of these factors are the same as those described in Table 3.2.

Table 3.4: Cost difference statistics (%) in the modified numerical study

<table>
<thead>
<tr>
<th></th>
<th>median</th>
<th>mean</th>
<th>standard deviation</th>
<th>max</th>
<th>min</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>1.113</td>
<td>1.469</td>
<td>1.314</td>
<td>10.966</td>
<td>8.07 × 10⁻⁶</td>
</tr>
</tbody>
</table>

We find that the inconsistency rate now increases to 7.05% of all examples. Once
again, we focus on all the comparisons where the cost and complexity were inconsistent,
and determine the cost difference between the two supply chains in each and every one
of those cases. See Table 3.4 for the summary statistics of these cost differences. As
expected, this new study, which deliberately creates uneven demand vectors, leads to larger
cost differences. For example, the maximum cost difference in the event of inconsistency
3.6 Complexity versus Cost When Choosing a Configuration

In Section 3.5 we studied the relationship between cost and complexity by comparing supply chains that have the same configuration but different demand vectors. We now employ complexity to compare two supply chains that have the same demand vector but different supply chain configurations. In particular, we will compare a prototypical modular supply chain with a prototypical non-modular supply chain. See Figure 3.7 for a depiction of the modular and non-modular supply chain configurations. The non-modular supply chain is a two-echelon supply chain where four suppliers serve the final assembler, who then puts together the parts produced by the suppliers. In comparison, in the modular assembly supply chain, there is a mid-echelon with two suppliers, each of which assembles...
Figure 3.7: Prototypical configurations for non-modular vs. modular assembly supply chains

parts produced by two suppliers in the upper echelon. The final assembler then receives modules from the mid-echelon suppliers and assembles these modules.

We assume all the nodes in the uppermost echelon of both the modular and non-modular supply chains provide the same number of variants, denoted by $\gamma$. Consequently, the final assemblers in both supply chains offer the same number of variants, $\gamma^4$, and we assume that their demand vectors are the same, since our goal is to compare the configurations only.

As for the cost model, recall that the cost coefficient of node $i$, which depends only on the unit underage and overage costs and the service level, has been defined as

$$C_i := (c_{io} + c_{iu})\alpha z(\alpha) - c_{iu}z(\alpha) + (c_{io} + c_{iu})\phi_N(z(\alpha)).$$

Here, we slightly modify this notation. Let $C^N_i$ denote the cost coefficient of node $i$ in the non-modular supply chain and $C^M_i$ the cost coefficient of node $i$ in the modular supply chain. (See Figure 3.7 for the labeling of the nodes.) For ease of exposition, we assume that all nodes in the same echelon have the same cost coefficient, i.e., $C^N_1 = C^N_2 = C^N_3 = C^N_4$, $C^M_1 = C^M_2 = C^M_3 = C^M_4$ and $C^M_5 = C^M_6$. Furthermore, to make a fair comparison between the two supply chains, we assume that the final assembler of both supply chains have the same cost coefficient, i.e., $C^M_7 = C^N_7$, as do the suppliers in the most upstream echelon, i.e., $C^N_1 = \ldots = C^N_4 = C^M_1 = \ldots = C^M_4$.

In this section we assume that the leadtime to replenish the inventory of a variant at node $i$ is $L_i$ periods, that is, the number of suppliers to node $i$ or, equivalently, the number of inputs that are assembled by node $i$. This assumption is meant to reflect the fact that the larger the number of inputs that go into a variant, the longer the time it takes to assemble
the variant. As a consequence of this assumption, the replenishment lead time of the final assembler is four periods in the non-modular assembly supply chain and two periods in the modular assembly supply chain. The leadtime reduction in the modular supply chain helps the final assembler reduce its inventory costs compared to the non-modular supply chain. On the flipside, however, there are two mid-echelon suppliers in the modular assembly supply chain, which do not exist in the non-modular assembly supply chain. These new suppliers inflate the total inventory cost of the modular assembly supply chain compared to the non-modular one. Hence, when using cost as the criterion, the trade-off in moving from a non-modular configuration to a modular one is the cost reduction achieved by the final assembler in the modular supply chain versus the additional costs created by two new suppliers.

Because the complexity criterion does not explicitly recognize the cost parameters, it is possible that the complexity criterion will lead to markedly different choices between two configurations. Nonetheless, the complexity criterion may be promising, because it leads to a similar trade-off as the cost criterion. The complexity of each node is weighted by the number of links to that node, meaning that the final assembler’s contribution to complexity is lower in the modular supply chain, but the modular supply chain’s complexity is inflated by the addition of two new suppliers in the mid-echelon.

Given that the complexity criterion follows a similar trade-off as the cost criterion, but does not take into account any of the cost information, it is not clear whether the cost and complexity criteria will yield similar results. The next proposition sheds some light on this question.

**Proposition 3.4** Consider the modular and non-modular assembly supply chains shown in Figure 3.7 and assume that the final assemblers of the two chains have the same demand vector and the demand shares are equal across all the variants of the final assembler. Then:

(a) **According to complexity criterion:** If the number of variants offered by a node in the most upstream echelon, $\gamma$, is two or more, then the modular assembly supply chain is better. If $\gamma = 1$, then the non-modular assembly supply chain is better.

---

3We observed in the last section that arc-based complexity performs better than node-based complexity. Hence, in this section the complexity measure we use is the arc-based complexity.
(b) **According to cost criterion:** There exists a threshold $t$ such that if the number of variants offered by a node in the most upstream echelon, $\gamma$, is greater than or equal to $t$, then the modular assembly supply chain is better. If $\gamma < t$, then the non-modular assembly supply chain is better.

The proposition shows that cost and complexity may lead to different choices between modular and non-modular configurations: The threshold $t$, which applies when the cost criterion is used, may be different from one, in which case the cost and complexity criteria may disagree. The upside of this proposition, however, is that the choice between non-modular and modular supply chains exhibits the same trend with respect to the number of variants, $\gamma$, regardless of whether cost or complexity is used: a larger number of variants favors the modular assembly supply chain.

Proposition 3.4 uses the assumption that the demand shares of all variants produced by the final assembler are equal. We next relax this assumption and analyze how the choice between two configurations depends on the demand share of a given variant under both the cost and complexity criteria. In particular, consider one of the variants produced by a node in the most upstream echelon, say, variant $V_1$ of node 1. Let us write the demand vector of node 1 as $(a_1(1-p), a_2(1-p), ..., a_{n-1}(1-p), p)$, where $p$ is the demand share of variant $V_1$ at node 1 and the remaining demand at this node is shared arbitrarily by the other variants produced by the node. Proposition 3.5 describes how the choice between modular and non-modular configurations depends on $p$.

**Proposition 3.5** Consider the modular and non-modular assembly supply chains shown in Figure 3.7 and assume that the final assemblers of the two chains have the same demand vector. Let $(a_1(1-p), a_2(1-p), ..., a_{n-1}(1-p), p)$ be the demand vector of node 1. Then:

(a) **According to complexity criterion:** One of the following is true:

(i) The non-modular supply chain is better for all $p \in [0, 1]$, or

(ii) The modular supply chain is better for all $p \in [0, 1]$, or

(iii) There exist $p_1$ and $p_2$ such that $0 \leq p_1 < p_2 \leq 1$ and the non-modular supply chain is better for $p \in [0, p_1)$, the modular supply chain is better for $p \in [p_1, p_2)$ and the non-modular supply chain is better for $p \in [p_2, 1]$. 

68
According to cost criterion: One of the following is true:

(i) The non-modular supply chain is better for all \( p \in [0, 1] \), or

(ii) The modular supply chain is better for all \( p \in [0, 1] \), or

(iii) There exist \( \hat{p}_1 \) and \( \hat{p}_2 \) such that \( 0 \leq \hat{p}_1 < \hat{p}_2 \leq 1 \) and the non-modular supply chain is better for \( p \in [0, \hat{p}_1) \), the modular supply chain is better for \( p \in [\hat{p}_1, \hat{p}_2) \) and the non-modular supply chain is better for \( p \in [\hat{p}_2, 1] \).

The first observation from the proposition is that cost and complexity may lead to different decisions depending on how similar or dissimilar \( p_1 \) and \( p_2 \) are to \( \hat{p}_1 \) and \( \hat{p}_2 \). Nonetheless, the proposition shows that the structural behavior of the modular versus non-modular configuration choice is the same under both cost and complexity. The intuition behind the result is as follows. If \( p \) is close to zero, the implication is that there is very little demand for the final assembler’s products that use this variant. Hence, it is almost as if those products do not exist, which is effectively equivalent to reducing the variety offered by the supply chain. Hence, the non-modular configuration is preferred. Likewise, when \( p \) is close to one, the demand shares of all other variants at supplier 1 decreases, thus resulting in a number of products with practically no demand at the final assembler. This again results in non-modular supply chain being preferred. However, when \( p \) is moderate, the variant does not depress the variety offered by the supply chain in any way, which motivates the use of modular configuration.

The previous propositions show that while cost and complexity may show similar behavior, they may lead to different configuration choices because the decisions depend on thresholds that may differ between cost and complexity. In particular, the thresholds for cost criterion depend on the cost coefficients of the nodes whereas complexity does not pay any attention to the cost coefficients. This negligence on complexity criterion’s part may cause significant inconsistencies when it comes to configuration choice. To further understand the effect of cost coefficients on the consistency between cost and complexity, we ask the following question: What can we say about the effect of the final assembler’s cost coefficient on the consistency between cost and complexity? The following proposition provides a partial answer to this question.
Proposition 3.6  Consider the modular and non-modular assembly supply chains shown in Figure 3.7 and, for a given set of parameters, suppose that both cost and complexity criteria favor the non-modular supply chain. Then:

(a) If the final assembler’s cost coefficient decreases, then cost and complexity continue to be consistent and both favor the non-modular supply chain.

(b) If the final assembler’s cost coefficient increases beyond a certain threshold, then cost begins to favor the modular supply chain and complexity continues to favor the non-modular supply chain, thereby creating an inconsistency between the two criteria.

One implication of the proposition is that whether cost and complexity agree depends very much on what the cost coefficients are. In our numerical examples, we have observed that the consistency is very sensitive to the cost coefficients. In fact, the consistency is so sensitive to the cost coefficient that cost and complexity may almost always agree for one set of cost coefficients and may almost never agree for another set of cost coefficients, regardless of where the other parameters are set such as number of variants and the demand vectors. Hence, one needs to be cautious when using complexity to make configuration choices.

3.7 Conclusion

In this paper we proposed a complexity measure for assembly supply chains, based on the concept of information entropy. This complexity measure takes into account factors such as the supply chain configuration, the level of variety offered by each node of the supply chain, and the demand split across all the variants offered by a node. We investigated the relationship between the complexity and the cost of an assembly supply chain. In particular, we showed that, when comparing assembly supply chains with the same configuration but different levels of product variety, the cost and the complexity are equivalent under certain conditions. Even when these conditions do not hold, our numerical study demonstrated that the cost and complexity criteria rank supply chains consistently in an overwhelming majority of cases. The agreement between the cost and complexity criteria was shown to be lesser when comparing assembly supply chains with the same level of product variety, but different configurations. Overall, we found that the complexity measure is a good proxy for
the cost of an assembly supply chain when evaluating alternative levels of product variety that will be delivered by a given supply chain configuration, but not so when evaluating alternative supply chain configurations to deliver a given level of product variety.

In Section 3.5, we observed that complexity is a good proxy for cost when making variety decisions, but our analysis in that section focused on a cost model where only inventory costs were accounted for. It is not difficult to extend the results to the case where transportation costs are also accounted for. In fact, as long as one is willing to assume that the transportation cost along an arc of the supply chain depends only on the total volume that travels along that arc, but not on the volumes of specific variants, all the results of Section 3.5 continue to hold. In order to extend the results to the case where manufacturing costs are also included in the supply chain cost, it would be sufficient to assume that the expected per-period manufacturing cost for variant \( v \) of node \( i \) is increasing and concave in the demand share of the variant, \( q_{iv} \). This assumption is not unreasonable: Under this assumption, given two nodes that differ only in terms of their demand vectors, the node that produces many medium-volume products will incur larger manufacturing costs compared to a node that produces a few high-volume products coupled with low-volume products. This outcome is reasonable because it captures the economies of scale that a node can enjoy by offering high-volume products.

Our analysis in Section 3.6 showed that, when using complexity to choose between modular and non-modular configurations, the consistency between cost and complexity is sensitive to the cost disparity between two echelons. This happens, because the complexity criterion uses no cost information at all. One could of course improve the consistency between cost and complexity by trying to reflect in the complexity definition the cost disparities that exist among the nodes of the supply chain. For example, one may want to weigh more heavily the complexity contribution of nodes whose unit holding and shortage costs are higher. While this may be an attractive modification, it may also beat the purpose of using the complexity criterion in the first place, because one important advantage of the complexity criterion is to absolve the decision maker of the need to rely on cost data.
Appendix

Appendix A: Derivation of Arc-based Complexity

*Random Experiment:* For each node in the supply chain, we form a pool of variants produced by that node, 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 variant from the pool of this arc’s end-node.

Let $L_i$ denote the number of suppliers connected to node $i$ in the supply chain, $i = 1, 2, \ldots, n$. Recall that we let $L_i = 1$ for any node $i$ in the uppermost echelon, corresponding to our convention that there is a virtual supplier that is linked to the nodes in the uppermost echelon. As discussed earlier, this convention allows us to capture the flows into the supply chain. Let us denote the virtual supplier as node 0. Then the total number of arcs in the supply chain equals to $K = \sum_{i=1}^{n} L_i$. Let $R_{ji}$ represent the arc starting from node $j = 0, 1, \ldots, n - 1$ and ending at node $i = 1, \ldots, n$. Then the probability of picking a certain arc $R_{ji}$ and then picking variant $v$ from the pool of end node $i$ is $\frac{q_{iv}}{K}$ (since we pick arc $R_{ji}$ with probability $\frac{1}{K}$ and variant $v$ of node $i$ with probability $q_{iv}$). Substituting probabilities of all possible outcomes back into Shannon’s information entropy equation (3.2), the information entropy of the random experiment yields our arc-based complexity:

$$H_A = -\sum_{i=1}^{n} \sum_{j \in S_i} \sum_{v=1}^{V_i} \frac{q_{iv}}{K} \log_2 \frac{q_{iv}}{K}$$  \hspace{1cm} (3.10)

For any node $i = 1, \ldots, n$,

$$-\sum_{j \in S_i} \sum_{v=1}^{V_i} \frac{q_{iv}}{K} \log_2 \frac{q_{iv}}{K} = -L_i \sum_{v=1}^{V_i} \frac{q_{iv}}{K} \log_2 \frac{q_{iv}}{K}.$$  \hspace{1cm} (3.11)

Therefore, equation (3.10) can be rewritten as

$$H_A = -\sum_{i=1}^{n} V_i \frac{q_{iv}}{K} \log_2 \frac{q_{iv}}{K}$$  \hspace{1cm} (3.11)
Appendix B: Proofs of Propositions

Throughout the appendix, we make frequent use of the majorization theory. Hence, we start with a definition of majorization.

**Definition 1** (Marshall and Olkin, 1979) For any real vector \( x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n \), let \( x_{(1)} \geq x_{(2)} \geq \ldots \geq x_{(n)} \) denote the components of \( x \) in non-increasing order.

**B1.1** For \( x, y \in \mathbb{R}^n \), we say \( x \) majorizes \( y \), written as \( x \succ y \) if

\[
\sum_{i=1}^{k} x(i) \geq \sum_{i=1}^{k} y(i), k = 1, \ldots, n-1 \text{ and } \sum_{i=1}^{n} x(i) = \sum_{i=1}^{n} y(i),
\]

**B1.2** \( x \succ y \) only if \( \sum g(x_i) \leq \sum g(y_i) \) for all continuous concave functions \( g : \mathbb{R} \rightarrow \mathbb{R} \).

**Proof of Proposition 3.1**

Given the demand shares are equal across all variants of the final assembler (indexed as node \( n \) according to our convention), the demand vector of supply chain \( k \)'s final assembler is \( q^n_k = \left( \frac{1}{V^n_k}, \frac{1}{V^n_k}, \ldots, \frac{1}{V^n_k} \right) \), where \( V^n_k \) is the number of variants offered by supply chain \( k \)'s final assembler and \( q^n_k \) is a vector of dimension \( V^n_k \). It now follows from (3.1) that all other nodes in supply chain \( k \) also have evenly-distributed demand vectors, i.e., \( q^i_k = \left( \frac{1}{V^i_k}, \frac{1}{V^i_k}, \ldots, \frac{1}{V^i_k} \right) \), \( i = 1, 2, \ldots, n \), where \( q^i_k \) is a vector of dimension \( V^i_k \).

First, consider two supply chains, indexed as 1 and 2, and suppose that supply chain 2's final assembler offers more variants than supply chain 1's, that is, \( V^2_n > V^1_n \). Then, each and every node in supply chain 2 offers at least as many variants as its counterpart in supply chain 1, that is, \( V^2_i \geq V^1_i \), \( i = 1, 2, \ldots, n \). To enable the use of majorization arguments when comparing the demand vectors of these two supply chains, we define, for \( i = 1, \ldots, n \), vectors \( x_i \) and \( y_i \), both of dimension \( V^2_i \), as follows:

\[
x_{iv} = q^1_{iv} = \frac{1}{V^1_i} \text{ for } v = 1, \ldots, V^1_i \text{ and } x_{iv} = 0 \text{ for } v = V^1_i + 1, \ldots, V^2_i,
\]

\[
y_{iv} = q^2_{iv} = \frac{1}{V^2_i} \text{ for } v = 1, \ldots, V^2_i.
\]
Using the definitions of vectors $\mathbf{x}_i$ and $\mathbf{y}_i$ and the definition of node-based complexity, given by (3.3), we can write

\[
H_N^1 = -\sum_{i=1}^{n} \sum_{v=1}^{V_i^2} \frac{x_{iv}}{n} \log_2 \frac{x_{iv}}{n},
\]

\[
H_N^2 = -\sum_{i=1}^{n} \sum_{v=1}^{V_i^2} \frac{y_{iv}}{n} \log_2 \frac{y_{iv}}{n}.
\]

Now, observe that $-x \log_2 x$ is concave in $x$. In addition, notice that the vector $\mathbf{x}_i$ majorizes the vector $\mathbf{y}_i$ for $i = 1, \ldots, n$. Thus, we can apply the majorization theorem to obtain $H_N^2 \geq H_N^1$.

Using the definition of arc-based complexity, given by (3.5), we can write

\[
H_A^1 = -\sum_{i=1}^{n} L_i \sum_{v=1}^{V_i^2} \frac{x_{iv}}{K} \log_2 \frac{x_{iv}}{K},
\]

\[
H_A^2 = -\sum_{i=1}^{n} L_i \sum_{v=1}^{V_i^2} \frac{y_{iv}}{K} \log_2 \frac{y_{iv}}{K}.
\]

Applying the majorization theorem yields $H_A^2 \geq H_A^1$.

Similarly, using the expression for supply chain cost $I$, given by (3.8), we can write

\[
I^1 = \sum_{i=1}^{n} C_i \sqrt{\lambda(l_i + 1)} \sum_{v=1}^{V_i^2} \sqrt{x_{iv}},
\]

\[
I^2 = \sum_{i=1}^{n} C_i \sqrt{\lambda(l_i + 1)} \sum_{v=1}^{V_i^2} \sqrt{y_{iv}}.
\]

Observing that $\sqrt{x}$ is concave in $x$ and applying the majorization theorem, we obtain $I^2 \geq I^1$.

In summary, given two supply chains whose demand vectors are evenly distributed, arc-based complexity, node-based complexity and cost rank these two supply chains in the same order. Because the three orderings are the same when comparing an arbitrary pair of supply chains, they will be the same when comparing an arbitrary number of supply chains.

\[
\text{Notice that if } x_{iv} = 0, \text{ then } \frac{x_{iv}}{n} \log_2 \frac{x_{iv}}{n} = 0 \text{ as well.}
\]
Proof of Proposition 3.2

Since all supply chains are identical in terms of the number of variants offered by each node, we drop the superscript $k$ from the number of variants offered by node $i$ of supply chain, and we write $V_i$ instead of $V_i^k$. Consider two supply chains, where the demand vectors of the final assemblers in supply chains 1 and 2 are, respectively, $q^1_n = (q^1_{n,1}, q^1_{n,2}, ..., q^1_{n,V_n})$ and $q^2_n = (q^2_{n,1}, q^2_{n,2}, ..., q^2_{n,V_n})$. As required by the proposition, suppose that: (i) $q^1_{n,1} \geq q^1_{n,2} = \ldots = q^1_{n,V_n}$ and (ii) $q^2_{n,1} \geq q^2_{n,2} = \ldots = q^2_{n,V_n}$. Furthermore, without loss of generality, index the supply chains 1 and 2 so that: (iii) $q^1_{n,1} \geq q^2_{n,1}$. Next, we will show that node-based complexity, arc-based complexity and cost rank these two supply chains in the same order.

To do so, we first prove that properties (i) through (iii) are satisfied for nodes $i = 1, \ldots, n-1$ as well.

For a given node $i = 1, \ldots, n-1$, to see why (i) and (ii) hold, observe that one could 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 shares of the variants in each subset add up to the demand share of a variant at node $i$. As a result, at node $i$ of supply chain $k$, there must be $V_i - 1$ variants that all have the same demand share and the remaining variant’s demand share is $q^k_{n,1} + \frac{(1-q^k_{n,1})}{V_n} [(V_n/V_i) - 1]$, which is larger than the others. Let us index this variant with the larger share as variant 1. Now, for a given node $i = 1, \ldots, n-1$, to see why (iii) holds, observe that

$$q^1_{n,1} - q^2_{n,1} = q^1_{n,1} + [(V_n/V_i) - 1] \frac{1-q^1_{n,1}}{V_n-1} - q^2_{n,1} - [(V_n/V_i) - 1] \frac{1-q^2_{n,1}}{V_n-1}$$

$$= (q^1_{n,1} - q^2_{n,1}) \left( 1 - \frac{(V_n/V_i) - 1}{V_n - 1} \right)$$

$$\geq 0$$

Now that we have shown properties (i) though (iii) hold for any node $i$, we can conclude that $q^1_i \succ q^2_i$ for node $i = 1, \ldots, n$. Applying the majorization theorem, we obtain $H^2_N \geq H^1_N$, $H^2_A \geq H^1_A$, and $I^2 \geq I^1$. Since the three orderings are the same for any arbitrary pair, it follows that the three orderings are the same for an arbitrary number of supply chains.
Proof of Proposition 3.3

For the purposes of this proof, we define $Q_i$ to be the set of suppliers in the uppermost echelon whose variants are used in the module produced by node $i$. For $a$ vectors $x_1, x_2, \ldots, x_a$ with dimensions, respectively, $m_1, m_2, \ldots, m_a$, define the operation $\Omega(x_1, \ldots, x_a)$ as the sorted component-wise multiplication of vectors $x_1, x_2, \ldots, x_a$, that is, the following vector with dimension $\prod_{j=1}^a m_j$:

$$(x_{11}x_{21}\cdots x_{a1}, x_{11}x_{21}\cdots x_{a2}, \ldots, x_{1,m_1}x_{2,m_2}\cdots x_{a,m_a}),$$

sorted in descending order. Using these definitions, notice that the demand vector of any given node $i$ can be written as $q_i = \Omega_{j\in Q_i} q_j$. This definition will be useful in the proof that follows.

We first prove the result for two supply chains. Suppose that there are $a$ suppliers in the most upstream echelon in both supply chains, denoted as nodes $1, \ldots, a$. Assume that, as required by the proposition, supply chains 1 and 2 have the same configuration and the demand vector of each node in the uppermost echelon of supply chain 1 majorizes its counter-part in supply chain 2, $q_1^s \succ q_2^s$, $s = 1, \ldots, a$. In this proof, we write the node-based complexity as a function of the demand vectors of the nodes in the uppermost echelon, that is, we write $H_N(q_1, \ldots, q_a)$ instead of $H_N(q_1, \ldots, q_n)$. (Note that this reduction of the argument list is possible because $q_{a+1}$ through $q_n$ can be recovered from $q_1, \ldots, q_a$, using the $\Omega$-operation defined above.) We will prove that, if $q_1^s \succ q_2^s$, $s = 1, \ldots, a$, then $H_N(q_1^1, \ldots, q_a^1) \leq H_N(q_1^2, \ldots, q_a^2)$, and $H_A(q_1^1, \ldots, q_a^1) \leq H_A(q_1^2, \ldots, q_a^2)$ and $I(q_1^1, \ldots, q_a^1) \leq I(q_1^2, \ldots, q_a^2)$. The proof is conducted in two steps.

Step 1: Suppose for now that the demand vectors of nodes 2 through $a$ are the same in both supply chains, but node 1’s demand vector in supply chain 1 majorizes the demand vector of node 1 in supply chain 2. That is, $q_1^1 \succ q_1^2$, $q_2^1 = q_2^2, \ldots, q_a^1 = q_a^2$. For notational convenience, let $q_j = q_j^1 = q_j^2, j = 2, \ldots, a$. We next prove that $q_i^1 \succ q_i^2$, $i = a + 1, \ldots, n$. 

76
If the variants provided by node \( i \) do not use any variant from node 1, i.e., \( 1 \notin Q_i \), then the demand vectors of node \( i \) in both supply chains 1 and 2 are the same, i.e., \( q^1_i = q^2_i \) for \( i = a + 1, \ldots, n \) such that \( 1 \notin Q_i \). Hence, \( q^1_i > q^2_i \) holds trivially for \( i = a + 1, \ldots, n \) such that \( 1 \notin Q_i \).

Consider now the case where one or more variants of node \( i \) use the variants from node 1, i.e., \( 1 \in Q_i \). We can write \( q^1_i = \Omega(q_1^1, \Omega_{j \in Q_i \setminus \{1\}} q_j) \) and \( q^2_i = \Omega(q_1^2, \Omega_{j \in Q_i \setminus \{1\}} q_j) \). Let \( A = q_1^1 \), \( B = q_1^2 \) and \( C = \Omega_{j \in Q_i \setminus \{1\}} q_j \) and apply Lemma 1 to conclude that \( q^1_i > q^2_i \).

Thus, \( q^1_i > q^2_i \) for \( i = 1, \ldots, n \). It now follows from the majorization theorem that \( H_N(q_1^2, q_2, \ldots, q_a) \geq H_N(q_1^1, q_2, \ldots, q_a) \), \( H_A(q_1^2, q_2, \ldots, q_a) \geq H_A(q_1^1, q_2, \ldots, q_a) \) and \( I(q_1^2, q_2, \ldots, q_a) \geq I(q_1^1, q_2, \ldots, q_a) \).

**Step 2:** It follows from Step 1 that, if \( q^1_i > q^2_i \) for \( i = 1, \ldots, a \), then

\[
H_N(q_1^2, q_2^2, \ldots, q_{a-1}^2, q_a^2) \geq H_N(q_1^1, q_2^1, \ldots, q_{a-1}^1, q_a^2)
\]

\[
\geq H_N(q_1^1, q_2^1, \ldots, q_{a-1}^1, q_a^2)
\]

\[
\geq \ldots \geq H_N(q_1^1, q_2^1, \ldots, q_{a-1}^1, q_a^2)
\]

\[
\geq H_N(q_1^1, q_2^1, \ldots, q_{a-1}^1, q_a^1)
\]

Hence, \( H_N(q_1^2, \ldots, q_a^2) \geq H_N(q_1^1, \ldots, q_a^1) \). Using the same line of arguments, we obtain

\[
H_A(q_1^2, \ldots, q_a^2) \geq H_A(q_1^1, \ldots, q_a^1) \)

\[
and \ I(q_1^2, \ldots, q_a^2) \geq I(q_1^1, \ldots, q_a^1).
\]

In Steps 1 and 2, we have proved that, given two supply chains that satisfy the conditions of the proposition, arc-based complexity, node-based complexity and cost rank these two supply chains in the same order. Because the three orderings are the same when comparing an arbitrary pair of supply chains, they will be the same when comparing an arbitrary number of supply chains.
Proof of Proposition 3.4

For the purposes of this proof, let $H_{A-N}$ and $H_{A-M}$ denote the arc-based complexity of the non-modular and modular supply chains, respectively. Likewise, let $I^N$ and $I^M$ denote the respective costs of the non-modular and modular supply chains. In addition, let $q^N_i$ and $q^M_i$ denote demand vector of node $i$ under non-modular and modular supply chains, respectively. Here the numbering of nodes follows the same convention introduced in Figure 3.7. Since the demand shares are equal across all the variants of the final assembler, we get the following demand vectors:

$$q^N_i = q^M_i = \left( \frac{1}{\gamma}, \frac{1}{\gamma}, \ldots, \frac{1}{\gamma} \right)_{1 \times \gamma}, \quad i = 1, 2, 3, 4 \quad (3.12)$$

$$q^M_5 = q^M_6 = \left( \frac{1}{\gamma^2}, \frac{1}{\gamma^2}, \ldots, \frac{1}{\gamma^2} \right)_{1 \times \gamma^2} \quad (3.13)$$

$$q^N_7 = q^M_7 = \left( \frac{1}{\gamma^4}, \frac{1}{\gamma^4}, \ldots, \frac{1}{\gamma^4} \right)_{1 \times \gamma^4} \quad (3.14)$$

Proof of (a). Using the complexity criterion: Recall that the arc-based complexity of an assembly supply chain is $\sum_{i=1}^n \sum_{v=1}^{V_i} -L_i \frac{q_i v}{K} \log_2 \frac{q_i v}{K}$. With some algebra, it can be checked that the arc-based complexity can also be written as

$$\log_2 K - \frac{1}{K} \sum_{i=1}^n L_i \sum_{v=1}^{V_i} q_{iv} \log_2 q_{iv}. \quad (3.15)$$

By substituting in (4.8) the expression for $q^N_i$, $i = 1, 2, 3, 4$ given by (3.12), and the expression for $q^N_7$, given by (3.14), and letting $K = 8$, $L_i = 1$, $i = 1, \ldots, 4$ and $L_7 = 4$, we obtain the following expression for the complexity of the non-modular assembly supply chain:

$$H_{A-N} = \frac{5}{2} \log_2 \gamma + \log_2 8.$$  

Similarly, by substituting in (4.8) the expression for $q^M_i$, $i = 1, 2, 3, 4$ given by (3.12), the expression for $q^M_i$, $i = 5, 6$ given by (3.13) and the expression for $q^M_7$, given by (3.14), and letting $K = 10$, $L_i = 1$, $i = 1, \ldots, 4$ and $L_5 = L_6 = L_7 = 2$, we obtain the following
expression for the complexity of the modular assembly supply chain:

\[ H^{A-M} = 2 \log_2 \gamma + \log_2 10. \]

The complexity difference between non-modular and modular supply chains is

\[ H^{A-N} - H^{A-M} = \frac{1}{2} \log_2 \gamma - 0.3219 \]

When \( \gamma = 1 \), \( H^{A-N} - H^{A-M} < 0 \), and the non-modular assembly supply chain is better. When \( \gamma \geq 2 \), \( H^{A-N} - H^{A-M} > 0 \), and the modular assembly supply chain is better.

**Proof of (b). Using the cost criterion:** Since we assume the leadtime of node \( i \) is \( L_i \), the number of inputs assembled at node \( i \). Therefore the cost of an assembly supply chain is

\[ I = \sum_{i=1}^{N} \sum_{v=1}^{V_i} I_{iv} = \sum_{i=1}^{n} \sum_{v=1}^{V_i} C_i \sqrt{(L_i + 1) \lambda q_{iv}}. \]  

(3.16)

Because we assume all the nodes in the same echelon have the same cost coefficient, we let, for notational convenience, \( C_I = C_i^N = C_i^M, i = 1, 2, 3, 4 \) and \( C_{II} = C_i^M, i = 5, 6 \) and \( C_{III} = C_i^N = C_i^M \).

By substituting in (3.16) the expression for \( q_i^N, i = 1, 2, 3, 4 \) given by (3.12), and the expression for \( q_i^M, i = 5, 6 \) given by (3.14), we obtain the following expression for the cost of the non-modular assembly supply chain:

\[ I^N = C_{III} \gamma^2 \sqrt{5\lambda} + 4C_I \sqrt{2\lambda \gamma}. \]

Similarly, by substituting in (3.16) the expression for \( q_i^M, i = 1, 2, 3, 4 \) given by (3.12), the expression for \( q_i^M, i = 5, 6 \) given by (3.13) and the expression for \( q_7^M \), given by (3.14), we obtain the following expression for the cost of the modular assembly supply chain:

\[ I^M = C_{III} \gamma^2 \sqrt{3\lambda} + 2C_{II} \gamma \sqrt{3\lambda} + 4C_I \sqrt{2\lambda \gamma}. \]
The cost difference between non-modular and modular supply chain is,

\[ I^N - I^M = -2\gamma C_{II} \sqrt{3} + \gamma^2 C_{III}(\sqrt{5} - \sqrt{3}). \]

Define \( t := \frac{2\sqrt{3} C_{II}}{(\sqrt{5} - \sqrt{3}) C_{III}} \). If \( \gamma \leq t \), \( I^N - I^M \leq 0 \) and non-modular assembly supply chain is better. If \( \gamma > t \), \( I^N - I^M > 0 \) and modular assembly supply chain is better.

**Proof of Proposition 3.5**

For the purposes of this proof, let \( H_{A-N} \) and \( H_{A-M} \) denote the arc-based complexity of the non-modular and modular supply chains, respectively. Likewise, let \( I^N \) and \( I^M \) denote the respective costs of the non-modular and modular supply chains. In addition, let \( q_i^N \) and \( q_i^M \) denote demand vector of node \( i \) under non-modular and modular supply chains, respectively. Here the numbering of nodes follows the same convention introduced in Figure 3.7. For notational convenience, let \( q_{iv} = q_i^N = q_i^M \), where \( i = 1, 2, 3, 4, 7, v = 1, 2, \ldots, V_i \).

Let \( q_{5v} = q_{5v}^M, v = 1, 2, \ldots, V_5 \) and \( q_{6v} = q_{6v}^M, v = 1, 2, \ldots, V_6 \).

**Proof of (a). Using the complexity criterion:** Recall that the arc-based complexity of an assembly supply chain is

\[ -\sum_{i=1}^n \sum_{v=1}^{V_i} L_i \frac{q_{iv}}{K} \log_2 \frac{q_{iv}}{K} = \log_2 K - \sum_{i=1}^n L_i \sum_{v=1}^{V_i} q_{iv} \log_2 q_{iv}. \quad (3.17) \]

By substituting \( K = 8, L_i = 1 \) for \( i = 1, \ldots, 4 \) and \( L_7 = 4 \) in (3.17), we obtain the following expression for the complexity of the non-modular assembly supply chain:

\[ H_{A-N} = \log_2 8 - \frac{1}{8} \sum_{i=1}^4 \sum_{v=1}^{\gamma} q_{iv} \log_2 q_{iv} - \frac{4}{8} \sum_{v=1}^{\gamma} q_{7v} \log_2 q_{7v} \]

Similarly, by substituting \( K = 10, L_i = 1 \) for \( i = 1, \ldots, 4 \), \( L_5 = L_6 = L_7 = 2 \) in (3.17), we obtain the following expression for the complexity of the modular assembly supply chain:

\[ H_{A-M} = \log_2 10 - \frac{1}{10} \sum_{i=1}^4 \sum_{v=1}^{\gamma} q_{iv} \log_2 q_{iv} - \frac{2}{10} \sum_{i=5}^6 \sum_{v=1}^{\gamma^2} q_{iv} \log_2 q_{iv} - \frac{2}{10} \sum_{v=1}^{\gamma^4} q_{7v} \log_2 q_{7v} \]
Therefore, the complexity difference between non-modular and modular assembly supply chains is

\[ H^{A-N} - H^{A-M} = \log_2 \frac{8}{10} - \frac{1}{40} \sum_{i=1}^{4} \sum_{v=1}^{\gamma} q_{iv} \log_2 q_{iv} + \frac{2}{10} \sum_{i=5}^{6} \sum_{v=1}^{\gamma} q_{iv} \log_2 q_{iv} - \frac{3}{10} \sum_{v=1}^{\gamma^4} q_{7v} \log_2 q_{7v} \]

Notice that \( q_{iv} \) depend on \( p \) only for \( i = 1, 5, 7 \). Hence, the term

\[ \kappa_1 := \log_2 \frac{8}{10} - \frac{1}{40} \sum_{i=1}^{\gamma} q_{1v} \log_2 q_{1v} + \frac{2}{10} \sum_{i=5}^{\gamma^2} q_{5v} \log_2 q_{5v} - \frac{3}{10} \sum_{v=1}^{\gamma^4} q_{7v} \log_2 q_{7v} + \kappa_1 \ (3.18) \]

in the difference \( H^{A-N} - H^{A-M} \) does not depend on \( p \), and we can rewrite \( H^{A-N} - H^{A-M} \) as

\[ H^{A-N} - H^{A-M} = -\frac{1}{40} \sum_{v=1}^{\gamma} q_{1v} \log_2 q_{1v} + \frac{2}{10} \sum_{v=1}^{\gamma^2} q_{5v} \log_2 q_{5v} - \frac{3}{10} \sum_{v=1}^{\gamma^4} q_{7v} \log_2 q_{7v} + \kappa_1 \ (3.18) \]

We know the following about the demand vectors \( q_1, q_5, \) and \( q_7 \):

\[ q_1 = (a_1(1-p), a_2(1-p), \ldots, a_{\gamma-1}(1-p), p)_{1 \times \gamma}, \text{ where } \sum_{i=1}^{\gamma-1} a_i = 1 \]

\[ q_5 = (a_1(1-p)q_{21}, \ldots, a_{\gamma-1}(1-p)q_{2\gamma}, pq_{21}, \ldots, pq_{2\gamma})_{1 \times \gamma^2} \]

\[ q_7 = (a_1(1-p)q_{21}q_{31}q_{41}, \ldots, a_{\gamma-1}(1-p)q_{2\gamma}q_{3\gamma}q_{4\gamma}, pq_{21}q_{31}q_{41}, \ldots, pq_{2\gamma}q_{3\gamma}q_{4\gamma})_{1 \times \gamma^4} \]

We substitute these demand vectors in (3.18) to obtain:

\[ H^{A-N} - H^{A-M} = \frac{1}{8} \left( p \sum_{i=1}^{\gamma-1} a_i \log_2 a_i - (1-p) \log_2(1-p) - p \log_2 p + \kappa_1 \right) \]

\[ -\frac{1}{10} \sum_{i=1}^{\gamma-1} a_i \log_2 a_i + \frac{2}{10} \sum_{j=1}^{\gamma} q_{2j} \log_2 q_{2j} - \frac{3}{10} \sum_{j=1}^{\gamma} \sum_{k=1}^{\gamma} q_{2j}q_{3k}q_{4l} \log_2 q_{2j}q_{3k}q_{4l} \]

Notice that the term

\[ \kappa_2 := \frac{1}{10} \sum_{i=1}^{\gamma-1} a_i \log_2 a_i + \frac{2}{10} \sum_{j=1}^{\gamma} q_{2j} \log_2 q_{2j} - \frac{3}{10} \sum_{j=1}^{\gamma} \sum_{k=1}^{\gamma} q_{2j}q_{3k}q_{4l} \log_2 q_{2j}q_{3k}q_{4l} \]
does not depend on $p$. We can rewrite the complexity difference between non-modular and modular assembly supply chains as:

$$H^{A-N} - H^{A-M} = \frac{1}{8} \left( p \sum_{i=1}^{\gamma-1} a_i \log_2 a_i - (1 - p) \log_2 (1 - p) - p \log_2 p \right) + \kappa_1 + \kappa_2$$

For ease of notation, let $R := \sum_{i=1}^{\gamma-1} a_i \log_2 a_i$ and $f(p) := \frac{1}{8} (pR - (1 - p) \log_2 (1 - p) - p \log_2 p)$. With these definitions, note that $H^{A-N} - H^{A-M} = f(p) + \kappa_1 + \kappa_2$. It is not difficult to check that $f(0) = 0$ and $f(1) < 0$. Furthermore:

$$f'(p) = \frac{1}{8} (R - \log_2 p + \log_2 (1 - p)),$$
$$f''(p) = -\frac{1}{8p(1-p)\ln 2} < 0 \text{ for } p \in (0,1).$$

Observe from above that $f(p)$ is concave in $p$. Define $p^*$ so that $f'(p^*) = 0$. One can check that $p^* = \frac{2R}{1+2\pi}$. Because $R \leq 0$, we observe $p^* = \frac{2R}{1+2\pi} \leq 1/2$. Therefore, the function $f(p)$ reaches its highest point for some $p \leq 1/2$. This observation, combined with $f(0) = 0$ and $f(1) < 0$, implies that the difference $H^{A-N} - H^{A-M} = f(p) + \kappa_1 + \kappa_2$ is concave in $p$ and reaches its highest point at some $p < 1$ and its lowest point at $p = 1$. Next, we use this information and we consider a series of cases to show all the possibilities about the sign of $H^{A-N} - H^{A-M}$ as a function of $p$.

**Case 1.** $f(1) + \kappa_1 + \kappa_2 \geq 0$: See Figure 3.8 (a) for an illustration of this case. Given that $H^{A-N} - H^{A-M}$ reaches its lowest point at $p = 1$, it follows that, in this case, $H^{A-N} - H^{A-M} \geq 0$ for all $p$. Therefore, the modular assembly supply chain is better for any $p$.

**Case 2.** $f(1) + \kappa_1 + \kappa_2 < 0$: In this case, we need to consider two subcases.

**Case 2(a).** $f(1) + \kappa_1 + \kappa_2 < 0$ and $f(0) + \kappa_1 + \kappa_2 \geq 0$: See Figure 3.8 (b) for an illustration of this case. In this case, the function $H^{A-N} - H^{A-M}$ starts out non-negative at $p = 0$ and ends up being negative at $p = 1$. Since the function $f(p)$ is strictly concave, there must exist $t \in [0,1]$ such that $H^{A-N} - H^{A-M} \geq 0$ for $p \leq t$ and $H^{A-N} - H^{A-M} < 0$ for $p > t$.

**Case 2(b).** $f(1) + \kappa_1 + \kappa_2 < 0$ and $f(0) + \kappa_1 + \kappa_2 < 0$: There are two further subcases to consider. If $f(p^*) + \kappa_1 + \kappa_2 \geq 0$, then there must exist $p_1$ and $p_2$ such that $H^{A-N} - H^{A-M}$ is
non-negative for all \( p \in [p_1, p_2] \) and negative elsewhere. (See Figure 3.8 (c) for an illustration of this case.) If \( f(p^*) + \kappa_1 + \kappa_2 < 0 \), then it must be that \( H^{A-N} - H^{A-M} \) is negative for all \( p \). (See Figure 3.8 (d) for an illustration of this case.)

**Proof of (b). Using the cost criterion:** We assume the leadtime of node \( i \) is \( L_i \), the number of inputs assembled at node \( i \). Then the cost of an assembly supply chain is

\[
I = \sum_{i=1}^{n} \sum_{v=1}^{V_i} I_{iv} = \sum_{i=1}^{n} \sum_{v=1}^{V_i} C_i \sqrt{(L_i + 1) \lambda q_{iv}}
\]  

(3.19)

By substituting \( L_i = 1 \) for \( i = 1, \ldots, 4 \) and \( L_7 = 4 \) in (3.19), we obtain the cost of the non-modular assembly supply chain:

\[
I^N = C_{III} \sqrt{5 \lambda} \sum_{v=1}^{V_7} \sqrt{q_{7v}} + C_{II} \sqrt{2 \lambda} \sum_{i=5}^{4} \sum_{v=1}^{V_i} \sqrt{q_{iv}}
\]

Similarly, by substituting \( L_i = 1 \) for \( i = 1, \ldots, 4 \), \( L_5 = L_6 = L_7 = 2 \) in (3.19), we obtain the cost of the modular assembly supply chain:

\[
I^M = C_{III} \sqrt{3 \lambda} \sum_{l=1}^{V_7} \sqrt{q_{7l}} + C_{II} \sqrt{3 \lambda} \sum_{i=5}^{5} \sum_{v=1}^{V_i} \sqrt{q_{iv}} + C_{I} \sqrt{2 \lambda} \sum_{i=1}^{4} \sum_{v=1}^{V_i} \sqrt{q_{iv}}
\]
Cost difference between non-modular and modular assembly supply chain is

\[ I^N - I^M = C_{III}(\sqrt{5} - \sqrt{3})\sqrt{\lambda} \sum_{v=1}^{V_7} \sqrt{q_{7v}} - C_{II} \sqrt{3\lambda} \left( \sum_{v=1}^{V_7} \sqrt{q_{5v}} + \sum_{v=1}^{V_6} \sqrt{q_{6v}} \right) \]  
(3.20)

Recall we have the following demand vectors:

\[ q_1 = (a_1(1-p), a_2(1-p), \ldots, a_{\gamma-1}(1-p), p)_{1\times\gamma}, \text{ where } \sum_{i=1}^{\gamma-1} a_i = 1 \]

\[ q_5 = (a_1(1-p)q_{21}, \ldots, a_{\gamma-1}(1-p)q_{2\gamma}, pq_{21}, \ldots, pq_{2\gamma})_{1\times\gamma^2} \]

\[ q_7 = (a_1(1-p)q_{21}q_{31}q_{41}, \ldots, a_{\gamma-1}(1-p)q_{2\gamma}q_{3\gamma}q_{4\gamma}, pq_{21}q_{31}q_{41}, \ldots, pq_{2\gamma}q_{3\gamma}q_{4\gamma})_{1\times\gamma^4} \]

By substituting the demand vector \( q_5 \) and \( q_7 \) back into (3.20), we can rewrite the cost difference between non-modular and modular assembly supply chains as

\[ I^N - I^M = S \cdot R \sqrt{1-p} + R \sqrt{p} - T \]

where \( S, R \) and \( T \), constants with respect to \( p \), are given by

\[ S = \sum_{i=1}^{\gamma-1} \sqrt{a_i}, \]

\[ R = C_{III}(\sqrt{5} - \sqrt{3})\sqrt{\lambda} \sum_{j=1}^{\gamma} \sum_{k=1}^{\gamma} \sum_{l=1}^{\gamma} \sqrt{q_{2j}q_{3k}q_{4l}} - C_{II} \sqrt{3\lambda} \sum_{j=1}^{\gamma} \sqrt{q_{2j}}, \]

\[ T = C_{II} \sqrt{3\lambda} \sum_{v=1}^{V_6} \sqrt{q_{6v}}. \]

Let \( g(p) = S \cdot R \sqrt{1-p} + R \sqrt{p} \). With this definition, notice that \( I^N - I^M = g(p) - T \).

Taking derivatives of \( G(p) \), we obtain:

\[ g'(p) = -\frac{SR}{\sqrt{1-p}} + \frac{R}{\sqrt{p}} \]

\[ g''(p) = -R \left( \frac{S}{4} (1-p)^{-3/2} + \frac{1}{4} p^{-3/2} \right) \]

Notice from above that \( G(p) \) is concave if \( R \geq 0 \) and convex if \( R < 0 \). Hence, we divide the proof into two cases depending on whether \( R \) is negative or non-negative.
(i) $R \geq 0$:

If $R \geq 0$, then $g''(p) \leq 0$, and $g(p)$ is concave. Let $p^*$ be the maximizer of $g(p)$. One can check that $p^* = \frac{R^2}{S^2 + R^2}$ and $p^* \in (0, 1)$. In addition, $g(0) = SR$ and $g(1) = R$. Because $R \geq 0$ and $S = \sum_{i=1}^{\gamma-1} \sqrt{a_i} \geq 1$, it follows that $g(0) = SR \geq g(1) = R$. Using these observations, we note that $g(p)$ is concave and reaches its highest point at $p^* < 1$ and its lowest point at $p = 1$. Keeping this in mind, we now consider a number of subcases, depending on the value of $T$ relative to $R \leq SR \leq g(p^*)$.

Case 1. $T \leq R$: See Figure 3.9 (a) for an illustration of this case. The lowest point of $I^N - I^M$, which occurs at $p = 1$, is given by $g(1) - T = R - T \geq 0$. Hence, $I^N - I^M \geq 0$ for all $p \in [0, 1]$, and the modular assembly supply chain is better for all $p \in [0, 1]$.

Case 2. $R < T \leq SR$: See Figure 3.9 (b) for an illustration of this case. The function $I^N - I^M$ is equal to $SR - T \geq 0$ at $p = 0$ and is equal to $R - T < 0$ at $p = 1$. Since the function $g(p)$ is strictly concave, there must exist $t \in [0, 1]$ such that $I^N - I^M \geq 0$ for $p \leq t$ and $I^N - I^M < 0$ for $p > t$.

Case 3. $R \leq SR < T \leq g(p^*)$: See Figure 3.9 (c) for an illustration of this case. The function $I^N - I^M$ is equal to $SR - T < 0$ at $p = 0$, reaches $g(p^*) - T \geq 0$ at its peak and
is equal to $R - T \leq 0$. In other words, $I^N - I^M$ starts negative, becomes positive and then again negative. Given $g(p)$ is concave, there must exist $p_1$ and $p_2$ such that $I^N - I^M$ is non-negative for all $p \in [p_1, p_2]$ and negative elsewhere.

**Case 4.** $R \leq SR \leq g(p^*) < T$: See Figure 3.9 (d) for an illustration of this case. The function $I^N - I^M$ is equal to $g(p^*) - T < 0$ at its peak. Therefore, $I^N - I^M < 0$ for all $p$, which implies that the non-modular assembly supply chain is better for any $p$.

(ii) $R < 0$:

If $R < 0$, then $g''(p) > 0$, and $g(p)$ is convex. Let $p^*$ be the minimizer of $g(p)$. One can check that $p^* = \frac{R^2}{S^2 + R^2}$ and $p^* \in (0, 1)$. In addition, $g(0) = SR$ and $g(1) = R$. Because $R < 0$ and $S = \sum_{i=1}^{n-1} \sqrt{\alpha_i} \geq 1$, it follows that $g(0) = SR \leq g(1) = R$. Using these observations, we note that $g(p)$ is convex and reaches its lowest point at $p^* < 1$ and its highest point at $p = 1$. Since $g(1) = R < 0$, then it follows that $g(p) < 0$ for all $p \in [0, 1]$. Hence, $I^N - I^M = g(p) - T < 0$ for all $p \in [0, 1]$ and the non-modular assembly chain has lower cost than modular assembly supply chain for any $p$.

Proof of Proposition 3.6

For the purposes of this proof, let $I^N$ and $I^M$ denote the respective costs of the non-modular and modular supply chains. In addition, let $q^N_i$ and $q^M_i$ denote demand vector of node $i$ under non-modular and modular supply chains, respectively. Here the numbering of nodes follows the same convention introduced in Figure 3.7. Notice that $q^N_i = q^M_i$ for node $i = 1, 2, 3, 4, 7$, because the supply chains are the same in terms of the final assembler’s demand vector. For notational convenience, let $q_{iv} = q^N_{iv} = q^M_{iv}$, where $i = 1, 2, 3, 4, 7$, $v = 1, 2, \ldots, V_i$. Let $q_{5v} = q^M_{5v}$, $l = 1, 2, \ldots, V_5$ and $q_{6v} = q^M_{6v}$, $l = 1, 2, \ldots, V_6$. Because we assume all the nodes in the same echelon have the same cost coefficient, we let, for notational convenience, $C_i = C^N_i = C^M_i$, $i = 1, 2, 3, 4$ and $C_{II} = C^M_{II}$, $i = 5, 6$ and $C_{III} = C^N_7 = C^M_7$. Recall that costs of an assembly supply chain as follows, if we assume the leadtime of node $i$ is $L_i$, the number of inputs assembled at node $i$.

$$I = \sum_{i=1}^{n} \sum_{v=1}^{V_i} I_{iv} = \sum_{i=1}^{n} \sum_{v=1}^{V_i} C_i \sqrt{(L_i + 1)\lambda q_{iv}}$$
By substituting $L_i = 1$ for $i = 1, \ldots, 4$ and $L_7 = 4$ in the above equation, we obtain the cost of the non-modular assembly supply chain:

$$I^N = C_I \sqrt{2\lambda} \sum_{i=1}^{4} \sum_{v=1}^{V_i} \sqrt{q_{iv}} + C_{III} \sqrt{5\lambda} \sum_{v=1}^{V_7} \sqrt{q_{7v}}$$

Similarly, by substituting $L_i = 1$ for $i = 1, \ldots, 4$, $L_5 = L_6 = L_7 = 2$ in the above equation, we obtain the cost of the modular assembly supply chain:

$$I^M = C_I \sqrt{2\lambda} \sum_{i=1}^{4} \sum_{v=1}^{V_i} \sqrt{q_{iv}} + C_{III} \sqrt{3\lambda} \sum_{i=5}^{5} \sum_{i=1}^{V_i} \sqrt{q_{iv}} + C_{II} \sqrt{3\lambda} \sum_{v=1}^{V_7} \sqrt{q_{7v}} + C_{III} \sqrt{3\lambda} \sum_{v=1}^{V_7} \sqrt{q_{7v}}$$

Cost difference between non-modular and modular assembly supply chain is

$$I^N - I^M = -C_{III} \sqrt{3\lambda} \left( \sum_{v=1}^{V_5} \sqrt{q_{5v}} + \sum_{v=1}^{V_6} \sqrt{q_{6v}} \right) + C_{III} (\sqrt{5} - \sqrt{3}) \sqrt{\lambda} \sum_{v=1}^{V_7} \sqrt{q_{7v}} \quad (3.21)$$

Observe from (3.21) that $I^N - I^M$ is an increasing function of $C_{III}$. Suppose that $I^N - I^M < 0$ at a given value of $C_{III}$, which means that the cost criterion favors the non-modular assembly supply chain. If we decrease the cost coefficient of the final assembler, $C_{III}$, we will continue to have $I^N - I^M < 0$ (because $I^N - I^M$ is a increasing function of $C_{III}$), so the cost criterion continues to favor the non-modular supply chain. Since the complexity of a supply chain does not change when $C_{III}$ changes, part (a) of the proposition follows.

On the other hand, if we increase the cost coefficient of the final assembler, $C_{III}$, $I^N - I^M$ will eventually exceed zero. Hence, there is a threshold $T$ such that $I^N - I^M \geq 0$ for when $C_{III} \geq T$, which means that the cost criterion favors the modular supply chain once $C_{III}$ exceeds a threshold. The complexity again does not depend on $C_{III}$. Hence, part (b) follows.

**Lemma 1** Suppose $m$-dimensional vectors $A = (a_1, a_2, \ldots, a_m)$ and $B = (b_1, b_2, \ldots, b_m)$ and $n$-dimensional vector $C = (c_1, c_2, \ldots, c_n)$ are all sorted in descending order. If $A > B$, then $\Omega(A, C) > \Omega(B, C)$ where the operation $\Omega(x, y)$ is the vector obtained by component-wise multiplication of vectors $x$ and $y$, sorted in descending order.
Proof of Lemma 1

Define \((x_1, x_2, \ldots, x_m)^\downarrow\) as the vector \((x_1, x_2, \ldots, x_m)\) sorted in descending order. We prove the result by induction on \(n\), the dimension of the vector \(C\). First, when \(n = 1\), in which case \(C = (c_1)\), we have:

\[
\Omega(A, C) = (a_1 c_1, a_2 c_1, \ldots, a_m c_1),
\]

\(3.22\)

\[
\Omega(B, C) = (b_1 c_1, b_2 c_1, \ldots, b_m c_1).
\]

\(3.23\)

Because \(A = (a_1, a_2, \ldots, a_m) \succ B = (b_1, b_2, \ldots, b_m)\), it follows that

\[
\Omega(A, C) = (a_1 c_1, a_2 c_1, \ldots, a_m c_1) \succ \Omega(B, C) = (b_1 c_1, b_2 c_1, \ldots, b_m c_1).
\]

Hence, the result holds when \(n = 1\). Suppose that if \(n = k - 1\), in which case \(C = (c_1, c_2, \ldots, c_{k-1})\), the result holds. That is:

\[
\Omega(A, C) = (a_1 c_1, a_2 c_1, \ldots, a_m c_1, a_1 c_{k-1}, \ldots, a_m c_{k-1})^\downarrow
\]

\[
\succ \Omega(B, C) = (b_1 c_1, b_2 c_1, \ldots, b_m c_1, b_1 c_{k-1}, \ldots, b_m c_{k-1})^\downarrow.
\]

In the remainder of the proof, define

\[
T = (t_1, t_2, \ldots, t_{m(k-1)}) := (a_1 c_1, a_2 c_1, \ldots, a_m c_1, a_1 c_{k-1}, \ldots, a_m c_{k-1})^\downarrow
\]

\[
S = (s_1, s_2, \ldots, s_{m(k-1)}) := (b_1 c_1, b_2 c_1, \ldots, b_m c_1, b_1 c_{k-1}, \ldots, b_m c_{k-1})^\downarrow.
\]

Notice that \(T \succ S\) by the induction assumption. To complete the induction, we will prove that if \(n = k\) and \(C = (c_1, c_2, \ldots, c_k)\), then \(\Omega(A, C) \succ \Omega(B, C)\). \(\Omega(A, C)\) is obtained by inserting the numbers \(a_1 c_k, \ldots, a_m c_k\) in descending order into the vector \(T\). \(\Omega(B, C)\) is obtained similarly by inserting the numbers \(b_1 c_k, \ldots, b_m c_k\) in descending order into the vector \(S\). To complete the induction, we use yet another induction, this time on the pairs of numbers inserted into vectors \(T\) and \(S\). To clarify, we will first show that, after inserting the first pair of numbers, \(a_1 c_k\) into \(T\) and \(b_1 c_k\) into \(S\), the resulting vectors \((T, a_1 c_k)^\downarrow\) and \((S, b_1 c_k)^\downarrow\) are such that \((T, a_1 c_k)^\downarrow \succ (S, b_1 c_k)^\downarrow\). We will then make the induction...
assumption that after inserting \( g - 1 \) pairs of numbers, we still have \((T, a_1c_k, \ldots, a_{g-1}c_k)^\downarrow > (S, b_1c_k, \ldots, b_{g-1}c_k)^\downarrow\). We will then show that, after inserting the \( g \)-th pair of numbers, \( a_gc_k \) and \( b_gc_k \), we have \((T, a_1c_k, \ldots, a_gc_k)^\downarrow > (S, b_1c_k, \ldots, b_gc_k)^\downarrow\). This will conclude the proof of both the inner and outer induction, concluding the proof of Lemma 1.

Suppose the first pair of numbers to be inserted, \( a_1c_k \) and \( b_1c_k \), are such that \( t_i \geq a_1c_k \geq t_{i+1} \) and \( s_j \geq b_1c_k \geq s_{j+1} \). Then, after inserting \( a_1c_k \) into \( T \) and \( b_1c_k \) into \( S \), and sorting the vectors, we get

\[
(T, a_1c_k)^\downarrow = (t_1, t_2, \ldots, t_i, a_1c_k, t_{i+1}, \ldots, t_{m(k-1)})
\]

\[
(S, b_1c_k)^\downarrow = (s_1, s_2, \ldots, s_j, b_1c_k, s_{j+1}, \ldots, s_{m(k-1)}).
\]

We next show that \((T, a_1c_k)^\downarrow > (S, b_1c_k)^\downarrow\). Consider two cases: \( i \leq j \) or \( i > j \).

**Case 1: \( i \leq j \)**

To see why \((T, a_1c_k)^\downarrow > (S, b_1c_k)^\downarrow\), note that:

(i) For \( z = 1, 2, \ldots, i \), we have \( \sum_{l=1}^z t_l \geq \sum_{l=1}^z s_l \) since \( T > S \).

(ii) For \( z = i+1, i+2, \ldots, j \), we have \( \sum_{l=1}^{z-1} t_l + a_1c_k \geq \sum_{l=1}^{z-1} t_l + t_z = \sum_{l=1}^{z-1} t_l + \sum_{l=1}^{z-1} s_l \), where the first inequality holds because (a) \( a_1c_k \geq t_{i+1} \) by assumption and (b) \( t_{i+1} \geq t_z \) since the vector \( T \) is sorted, and the second inequality holds because \( T > S \).

(iii) For \( z = j+1, j+2, \ldots, m(k - 1) + 1 \), \( \sum_{l=1}^{z-1} t_l + a_1c_k \geq \sum_{l=1}^{z-1} s_l + b_1c_k \) because (a) \( T > S \) implies that \( \sum_{l=1}^{z-1} t_l \geq \sum_{l=1}^{z-1} s_l \), and (b) \( A > B \) implies that \( a_1c_k \geq b_1c_k \).

**Case 2: \( i > j \)**

To see why \((T, a_1c_k)^\downarrow > (S, b_1c_k)^\downarrow\), note that:

(i) For \( z = 1, 2, \ldots, j \), we have \( \sum_{l=1}^z t_l \geq \sum_{l=1}^z s_l \) since \( T > S \).

(ii) For \( z = j+1, j+2, \ldots, i \), we have \( \sum_{l=1}^z t_l = \sum_{l=1}^{z-1} t_l + t_z \geq \sum_{l=1}^{z-1} s_l + a_1c_k \geq \sum_{l=1}^{z-1} s_l + b_1c_k \). The first inequality holds because \( T > S \) and \( t_i \geq a_1c_k \), and the second inequality holds because \( a_1c_k \geq b_1c_k \) from \( A > B \).

(iii) For \( z = i+1, i+2, \ldots, m(k - 1) + 1 \), \( \sum_{l=1}^{z-1} t_l + a_1c_k \geq \sum_{l=1}^{z-1} s_l + b_1c_k \) because (a) \( T > S \) implies that \( \sum_{l=1}^{z-1} t_l \geq \sum_{l=1}^{z-1} s_l \), and (b) \( A > B \) implies that \( a_1c_k \geq b_1c_k \).

Hence, we have shown that after inserting the first pair of numbers, \( a_1c_k \) and \( b_1c_k \), it is true that \((T, a_1c_k)^\downarrow > (S, b_1c_k)^\downarrow\). We now make the induction assumption that, after insert-
ing $a_{g-1}c_k$ and $b_{g-1}c_k$, it is true that $(T, a_1c_k, a_2c_k, \ldots, a_{g-1}c_k)^\updownarrow \succ (S, b_1c_k, b_2c_k, \ldots, b_{g-1}c_k)^\updownarrow$. For the remainder of the proof, define

$$T' = (t'_1, t'_2, \ldots, t'_{m-(k-1)+(g-1)}) := (T, a_1c_k, a_2c_k, \ldots, a_{g-1}c_k)^\updownarrow$$

$$S' = (s'_1, s'_2, \ldots, s'_{m-(k-1)+(g-1)}) := (S, b_1c_k, b_2c_k, \ldots, b_{g-1}c_k)^\updownarrow$$

Thus, the induction assumption can be written as $T' \succ S'$. To complete the induction, we prove that after inserting $a'gc_k$ and $b'gc_k$ into $T'$ and $S'$, it is true that

$$(T', a'gc_k)^\updownarrow = (T, a_1c_k, a_2c_k, \ldots, a_{g-1}c_k, a'gc_k)^\updownarrow$$

$$\succ (S', b'gc_k)^\updownarrow = (S, b_1c_k, b_2c_k, \ldots, b_{g-1}c_k, b'gc_k)^\updownarrow$$

Suppose $t'_i \geq a'gc_k \geq t'_{i+1}$ and $s'_j \geq b'gc_k \geq s'_{j+1}$. Then:

$$(T', a'gc_k)^\updownarrow = (T, a_1c_k, a_2c_k, \ldots, a_{g-1}c_k, a'gc_k)^\updownarrow = \left( t'_1, t'_2, \ldots, t'_i, a'gc_k, t'_{i+1}, \ldots, t'_{m-(k-1)+(g-1)} \right)$$

(3.24)

$$(S', b'gc_k)^\updownarrow = (S, b_1c_k, b_2c_k, \ldots, b_{g-1}c_k, b'gc_k)^\updownarrow = \left( s'_1, s'_2, \ldots, s'_j, b'gc_k, s'_{j+1}, \ldots, s'_{m-(k-1)+(g-1)} \right)$$

(3.25)

We will consider two cases: $i \leq j$, and $i > j$.

**Case 1: $i \leq j$**

To see why $(T', a'gc_k)^\updownarrow \succ (S', b'gc_k)^\updownarrow$, note that:

(i) For $z = 1, 2, \ldots, i$, we have $\sum_{l=1}^z t'_l \geq \sum_{l=1}^z s'_l$ since $T' \succ S'$.

(ii) For $z = i + 1, i + 2, \ldots, j$, we have $\sum_{l=1}^{z-1} t'_l + a'gc_k \geq \sum_{l=1}^{z-1} t'_l + t'_z = \sum_{l=1}^{z} t'_l \geq \sum_{l=1}^{z} s'_l$, where the first inequality holds because (a) $a'gc_k \geq t'_{i+1}$ by assumption and (b) the vector $T'$ is sorted and, hence, $t'_{i+1} \geq t'_z$, and the second inequality holds because $T' \succ S'$.

(iii) For $z = j + 1, j + 2, \ldots, m(k-1) + g$, the set $\{t'_1, \ldots, t'_{z-1}, a'gc_k\}$ can be divided into two disjoint subsets, $\{t'_1, \ldots, t'_{z-1}\} = \{a_1c_k, \ldots, a'gc_k\}$ (see (3.24)). Similarly, the set $\{s'_1, \ldots, s'_{z-1}, b'gc_k\}$ can also be divided into two disjoint subsets, $\{s'_1, \ldots, s'_{z-1}\} = \{b_1c_k, \ldots, b'gc_k\}$ (see (3.25)). Now, we obtain $\sum_{l=1}^{z-1} t'_l + a'gc_k = \sum_{l=1}^{z-1} t_l + \sum_{u=1}^{g} a_uc_k \geq$
\[ \sum_{l=1}^{z-g} t_l + \sum_{u=1}^{g} b_u c_k = \sum_{l=1}^{z-1} s_l' + b_g c_k, \] where the inequality holds because (a) \( \sum_{l=1}^{z-g} t_l \geq \sum_{l=1}^{z-1} s_l' \) by \( T > S \) and (b) \( \sum_{u=1}^{g} a_u c_k \geq \sum_{u=1}^{g} b_u c_k \) by \( A > B \).

**Case 2: i > j**

To see why \( (T', a_g c_k)^i > (S', b_g c_k)^i \), note that:

(i) For \( z = 1, 2, \ldots, j \), we have \( \sum_{l=1}^{i} t_l' \geq \sum_{l=1}^{z} s_l' \) since \( T' > S' \).

(ii) For \( z = j + 1, j + 2, \ldots, i \), the set \( \{t_1', \ldots, t_z'\} \) can be divided into two disjoint subsets, \( \{t_1, \ldots, t_{z-y}\} \) and \( \{a_1 c_k, \ldots, a_g c_k\} \), where \( y \leq g \). Then,

\[
\sum_{l=1}^{z} t_l' = \sum_{l=1}^{z-y} t_l + \sum_{u=1}^{y} a_u c_k = \sum_{l=1}^{z-g} t_l + \sum_{u=1}^{g} a_u c_k = \sum_{u=g+1}^{y} a_u c_k + \sum_{u=1}^{g} a_u c_k = \sum_{l=1}^{z-g} t_l + \sum_{u=g+1}^{y} a_u c_k \geq \sum_{u=g+1}^{y} a_u c_k + \sum_{u=1}^{g} a_u c_k = \sum_{l=1}^{z-g} s_l \text{ by } T > S \text{ and } (b) \sum_{u=1}^{g} a_u c_k \geq \sum_{u=1}^{g} b_u c_k \text{ by } A > B. \tag{3.26}
\]

where the inequality holds because \( T' \) is sorted in descending order and \( t_{z-y} \geq a_{y+1} c_k \). Now, the set \( \{s_1', \ldots, s_{z-1}, b_g c_k\} \) can also be divided into two disjoint subsets, \( \{s_1, \ldots, s_{z-g}\} \) and \( \{b_1 c_k, \ldots, b_g c_k\} \) (see (3.25)). We now note that \( \sum_{l=1}^{z-g} t_l + \sum_{u=1}^{g} a_u c_k \geq \sum_{l=1}^{z-1} s_l' + b_g c_k, \) where the first inequality follows from (3.26) and the second inequality holds because (a) \( \sum_{l=1}^{z-g} t_l \geq \sum_{l=1}^{z-1} s_l \) by \( T > S \) and (b) \( \sum_{u=1}^{g} a_u c_k \geq \sum_{u=1}^{g} b_u c_k \) by \( A > B \).

(iii) For \( z = i + 1, i + 2, \ldots, m(k-1) + g \), the set \( \{t_1', \ldots, t_{z-1}, a_g c_k\} \) can be divided into two disjoint subsets, \( \{t_1, \ldots, t_{z-g}\} \) and \( \{a_1 c_k, \ldots, a_g c_k\} \) (see (3.24)). Similarly, the set \( \{s_1', \ldots, s_{z-1}, b_g c_k\} \) can also be divided into two disjoint subsets, \( \{s_1, \ldots, s_{z-g}\} \) and \( \{b_1 c_k, \ldots, b_g c_k\} \) (see (3.25)). Now, we obtain \( \sum_{l=1}^{z-1} t_l' + a_g c_k = \sum_{l=1}^{z-g} t_l + \sum_{u=1}^{g} a_u c_k \geq \sum_{l=1}^{z-g} s_l + \sum_{u=1}^{g} b_u c_k = \sum_{l=1}^{z-1} s_l' + b_g c_k, \) where the inequality holds because (a) \( \sum_{l=1}^{z-g} t_l \geq \sum_{l=1}^{z-1} s_l \) by \( T > S \) and (b) \( \sum_{u=1}^{g} a_u c_k \geq \sum_{u=1}^{g} b_u c_k \) by \( A > B \).

Thus, we have proven the inner induction, which then proves the outer induction and concludes the proof.
### Appendix C: DOE Tables

Table 3.5: The percentage of inconsistencies for each of the 32 value combinations of the five factors shown in Table 3.1. The possible values for each factor is as shown in Table 3.1. For each combination, three replications are run.

<table>
<thead>
<tr>
<th>Run</th>
<th>Complexity</th>
<th>Num of Suppliers</th>
<th>Num of Variants</th>
<th>Cost Disparity</th>
<th>Demand Evenness</th>
<th>Inconsistency Rate ( %)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Factor</td>
<td>Definition</td>
<td></td>
<td></td>
<td></td>
<td>Rep 1</td>
</tr>
<tr>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2.31</td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2.82</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>3.24</td>
</tr>
<tr>
<td>4</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>4.59</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>3.16</td>
</tr>
<tr>
<td>6</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>3.51</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>3.54</td>
</tr>
<tr>
<td>8</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>4.27</td>
</tr>
<tr>
<td>9</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>2.43</td>
</tr>
<tr>
<td>10</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>3.26</td>
</tr>
<tr>
<td>11</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>3.74</td>
</tr>
<tr>
<td>12</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>5.05</td>
</tr>
<tr>
<td>13</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>3.8</td>
</tr>
<tr>
<td>14</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>4.26</td>
</tr>
<tr>
<td>15</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>3.66</td>
</tr>
<tr>
<td>16</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>4.35</td>
</tr>
<tr>
<td>17</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>0.5</td>
</tr>
<tr>
<td>18</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>1.1</td>
</tr>
<tr>
<td>19</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>0.8</td>
</tr>
<tr>
<td>20</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>1.77</td>
</tr>
<tr>
<td>21</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>0.64</td>
</tr>
<tr>
<td>22</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>0.92</td>
</tr>
<tr>
<td>23</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>0.56</td>
</tr>
<tr>
<td>24</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>0.88</td>
</tr>
<tr>
<td>25</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>0.86</td>
</tr>
<tr>
<td>26</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>1.5</td>
</tr>
<tr>
<td>27</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>0.86</td>
</tr>
<tr>
<td>28</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>1.93</td>
</tr>
<tr>
<td>29</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>0.82</td>
</tr>
<tr>
<td>30</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>1.28</td>
</tr>
<tr>
<td>31</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>0.49</td>
</tr>
<tr>
<td>32</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>0.86</td>
</tr>
</tbody>
</table>
Table 3.6: The percentage of inconsistencies for each of the 32 value combinations of the five factors shown in Table 3.1. The possible values for each factor is as shown in Table 3.1, except that the number of suppliers is now four or five (as opposed two or three). For each combination, three replications are run. Notice that the number of suppliers is no longer a statistically significant effect.

<table>
<thead>
<tr>
<th>Run</th>
<th>Factor</th>
<th>Inconsistency Rate (%)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Complexity Num of Num of Cost Demand</td>
<td>Rep Rep Rep</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Definition Suppliers Variants Disparity Evenness</td>
<td>1 2 3</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>- - - - -</td>
<td>3.78 3.75 3.61</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>+ - - - -</td>
<td>5.19 5.19 5.11</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>- + - - -</td>
<td>3.87 3.7 4.14</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>+ - + - -</td>
<td>5.82 5.42 5.93</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>- - + - -</td>
<td>3.71 3.67 3.81</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>+ - + - -</td>
<td>4.56 4.27 4.7</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>- + + - -</td>
<td>3.67 3.72 3.58</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>+ + + - -</td>
<td>4.55 4.66 4.38</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>- - - + +</td>
<td>4.36 4.01 3.95</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>+ - - + +</td>
<td>5.66 5.42 5.52</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>- + - + +</td>
<td>3.92 3.66 4.07</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>+ + - - +</td>
<td>5.71 5.46 5.92</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>- - - + +</td>
<td>3.76 3.48 3.78</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>+ - - + +</td>
<td>4.69 4.34 4.44</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>- + - + +</td>
<td>3.45 3.65 3.8</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>+ + - - +</td>
<td>4.33 4.46 4.78</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>- - - - +</td>
<td>0.86 0.62 0.82</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>+ - - - +</td>
<td>1.8 1.47 1.75</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>- + - - +</td>
<td>0.56 0.54 0.57</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>+ - - - +</td>
<td>1.42 1.36 1.3</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>- - + - +</td>
<td>0.57 0.61 0.57</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>+ - + - +</td>
<td>0.77 0.69 0.61</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>- + + - +</td>
<td>0.51 0.57 0.6</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>+ + + - +</td>
<td>0.62 0.75 0.8</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>- - - + +</td>
<td>0.76 0.79 0.69</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>+ - - + +</td>
<td>1.54 1.72 1.53</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>- + - + +</td>
<td>0.58 0.67 0.63</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>+ + - + +</td>
<td>1.33 1.31 1.38</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>- - + + +</td>
<td>0.53 0.63 0.64</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>+ - + + +</td>
<td>0.83 0.85 0.9</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>- + + + +</td>
<td>0.65 0.49 0.57</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>+ + + + +</td>
<td>0.8 0.73 0.68</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.7: Estimated effects, t-Statistics and p-Value (%) of the experiment shown in 3.6. A positive (negative) effect implies that when the factor’s value changes from ‘−’ to ‘+’, the inconsistency rate increases (decreases).

<table>
<thead>
<tr>
<th>Term</th>
<th>Effect</th>
<th>t-Statistics</th>
<th>p Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>186.28</td>
<td>&lt; 0.0001</td>
<td></td>
</tr>
<tr>
<td>Complexity definition</td>
<td>0.865</td>
<td>30.53</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>Number of suppliers</td>
<td>-0.026</td>
<td>-0.91</td>
<td>0.365</td>
</tr>
<tr>
<td>Number of variants</td>
<td>-0.520</td>
<td>-18.35</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>Cost disparity between two echelons</td>
<td>0.048</td>
<td>1.71</td>
<td>0.093</td>
</tr>
<tr>
<td>Evenness of the demand vector</td>
<td>-3.530</td>
<td>-124.57</td>
<td>&lt; 0.0001</td>
</tr>
</tbody>
</table>
Table 3.8: The percentage of inconsistencies for each of the 32 value combinations of the five factors shown in Table 3.1. The possible values for each factor is as shown in Table 3.1, except that the cost of the downstream echelon is now four or five times as large as the sum of cost coefficients of the nodes in the upstream echelon (as opposed to being the same as the sum or twice as large). For each combination, three replications are run. Notice that the cost disparity is no longer a statistically significant effect.

<table>
<thead>
<tr>
<th>Run</th>
<th>Complexity Definition</th>
<th>Num of Suppliers</th>
<th>Num of Variants</th>
<th>Cost Disparity</th>
<th>Evenness</th>
<th>Inconsistency Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Rep 1  Rep 2  Rep 3</td>
</tr>
<tr>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2.92 2.64 2.93</td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>3.45 3.21 3.46</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>3.59 4.19 3.59</td>
</tr>
<tr>
<td>4</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>4.94 5.51 4.66</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>3.48 3.88 3.51</td>
</tr>
<tr>
<td>6</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>3.87 4.38 3.83</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>3.71 3.73 3.64</td>
</tr>
<tr>
<td>8</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>4.44 4.42 4.17</td>
</tr>
<tr>
<td>9</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>2.77 2.82 2.81</td>
</tr>
<tr>
<td>10</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>3.6 3.41 3.42</td>
</tr>
<tr>
<td>11</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>3.92 4.18 3.97</td>
</tr>
<tr>
<td>12</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>5.23 5.39 5.09</td>
</tr>
<tr>
<td>13</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>4.02 3.93 3.77</td>
</tr>
<tr>
<td>14</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>4.48 4.38 4.02</td>
</tr>
<tr>
<td>15</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>3.73 3.66 4.27</td>
</tr>
<tr>
<td>16</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>4.42 4.45 4.99</td>
</tr>
<tr>
<td>17</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>0.89 0.95 0.81</td>
</tr>
<tr>
<td>18</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>1.49 1.45 1.38</td>
</tr>
<tr>
<td>19</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>0.97 1.04 1.16</td>
</tr>
<tr>
<td>20</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>1.94 1.91 2.05</td>
</tr>
<tr>
<td>21</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>0.88 0.99 0.9</td>
</tr>
<tr>
<td>22</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>1.2 1.49 1.22</td>
</tr>
<tr>
<td>23</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>0.63 0.77 0.5</td>
</tr>
<tr>
<td>24</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>0.97 0.97 0.85</td>
</tr>
<tr>
<td>25</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>1.05 0.85 0.94</td>
</tr>
<tr>
<td>26</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>1.69 1.37 1.62</td>
</tr>
<tr>
<td>27</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>0.91 0.95 0.9</td>
</tr>
<tr>
<td>28</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>1.98 1.79 1.92</td>
</tr>
<tr>
<td>29</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>0.98 0.77 0.77</td>
</tr>
<tr>
<td>30</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>1.44 1.31 1.14</td>
</tr>
<tr>
<td>31</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>0.5 0.76 0.69</td>
</tr>
<tr>
<td>32</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>0.87 1.01 0.99</td>
</tr>
</tbody>
</table>

Table 3.9: Estimated effects, t-Statistics and p-Value (%) of the experiment shown in Table 3.8. A positive (negative) effect implies that when the factor’s value changes from ‘-’ to ‘+’, the inconsistency rate increases (decreases).

<table>
<thead>
<tr>
<th>Term</th>
<th>Effect</th>
<th>t-Statistics</th>
<th>p Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>137.13</td>
<td>&lt; 0.0001</td>
<td></td>
</tr>
<tr>
<td>Complexity definition</td>
<td>0.647</td>
<td>17.49</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>Number of suppliers</td>
<td>0.382</td>
<td>10.33</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>Number of variants</td>
<td>-0.082</td>
<td>-2.21</td>
<td>0.030</td>
</tr>
<tr>
<td>Cost disparity between two echelons</td>
<td>0.091</td>
<td>2.46</td>
<td>0.017</td>
</tr>
<tr>
<td>Evenness of the demand vector</td>
<td>-2.797</td>
<td>-75.62</td>
<td>&lt; 0.0001</td>
</tr>
</tbody>
</table>
Table 3.10: The percentage of inconsistencies for each of the 32 value combinations of the five factors shown in Table 3.1. The possible values for each factor is as shown in Table 3.1, except that the number of suppliers is fixed at two and the factor of the number of suppliers is replaced with number of echelons (number of echelons could either be two or three). Notice that number of echelons is a statistically significant effect.

<table>
<thead>
<tr>
<th>Run</th>
<th>Complexity</th>
<th>Num of Echelon</th>
<th>Cost Variants</th>
<th>Demand Disparity</th>
<th>Evenness</th>
<th>Inconsistency Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Definition</td>
<td>Rep</td>
<td>Rep</td>
<td>Rep</td>
<td></td>
<td>1  2  3</td>
</tr>
<tr>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1.97 2.1 2.14</td>
</tr>
<tr>
<td>2</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2.59 2.64 2.76</td>
</tr>
<tr>
<td>3</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>3.99 3.61 3.81</td>
</tr>
<tr>
<td>4</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>4.59 4.16 4.42</td>
</tr>
<tr>
<td>5</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>3.13 3.59 3.12</td>
</tr>
<tr>
<td>6</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>3.63 3.96 3.48</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>1.42 1.51 1.58</td>
</tr>
<tr>
<td>8</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>1.61 1.68 1.92</td>
</tr>
<tr>
<td>9</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>2.77 2.49 2.65</td>
</tr>
<tr>
<td>10</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>3.42 3.14 3.37</td>
</tr>
<tr>
<td>11</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>4.35 4.66 4.27</td>
</tr>
<tr>
<td>12</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>4.92 5.31 4.74</td>
</tr>
<tr>
<td>13</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>3.65 3.45 3.66</td>
</tr>
<tr>
<td>14</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>4.12 3.99 4.05</td>
</tr>
<tr>
<td>15</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>2.94 3.27 3.23</td>
</tr>
<tr>
<td>16</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>3.18 3.67 3.43</td>
</tr>
<tr>
<td>17</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>0.49 0.49 0.53</td>
</tr>
<tr>
<td>18</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1.01 1.05 1.32</td>
</tr>
<tr>
<td>19</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.8 0.87 1</td>
</tr>
<tr>
<td>20</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>1.18 1.13 1.23</td>
</tr>
<tr>
<td>21</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>0.69 0.52 0.52</td>
</tr>
<tr>
<td>22</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>1.03 0.89 0.9</td>
</tr>
<tr>
<td>23</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>0.7 0.76 0.8</td>
</tr>
<tr>
<td>24</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>0.67 0.71 0.76</td>
</tr>
<tr>
<td>25</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>0.98 0.64 0.77</td>
</tr>
<tr>
<td>26</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>1.58 1.34 1.4</td>
</tr>
<tr>
<td>27</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>1.01 1.21 1.1</td>
</tr>
<tr>
<td>28</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>1.37 1.51 1.29</td>
</tr>
<tr>
<td>29</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>0.74 0.67 0.71</td>
</tr>
<tr>
<td>30</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>1.07 1.05 1.17</td>
</tr>
<tr>
<td>31</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>1.07 1.1 1.07</td>
</tr>
<tr>
<td>32</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>1.13 1.11 1.06</td>
</tr>
</tbody>
</table>

Table 3.11: Estimated effects, t-Statistics and p-Value (%) of the experiment shown in Table 3.10. A positive (negative) effect implies that when the factor’s value changes from ‘-’ to ‘+’, the inconsistency rate increases (decreases).

<table>
<thead>
<tr>
<th>Term</th>
<th>Effect</th>
<th>t-Statistics</th>
<th>p Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>143.02</td>
<td>&lt; 0.0001</td>
<td></td>
</tr>
<tr>
<td>Complexity definition</td>
<td>0.399</td>
<td>13.40</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>Number of echelons</td>
<td>0.198</td>
<td>6.64</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>Number of variants</td>
<td>-0.333</td>
<td>-11.20</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>Cost disparity between two echelons</td>
<td>0.530</td>
<td>17.79</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>Evenness of the demand vector</td>
<td>-2.332</td>
<td>-78.35</td>
<td>&lt; 0.0001</td>
</tr>
</tbody>
</table>
BIBLIOGRAPHY


CHAPTER 4

A COMPLEXITY MODEL FOR ASSEMBLY SUPPLY CHAINS AND ITS APPLICATION TO CONFIGURATION DESIGN

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, i.e., the demand share of one particular variant is bigger than the demand share of others, and 2) demand shares are equal across all variants at the final assembler. It is shown that in the first scenario, 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.

4.1 Introduction

In order to face the fierce competition in today’s market, manufacturers are motivated to provide high product variety and highly customized products. Modular product designs have been recognized as a key enabler for product variety. A modular design decomposes the product into several modules with standard interfaces and each module has a certain number of different variants. High product variety can be achieved through the combinational assembly of the variants from different modules while each module still keeps the high production volume, through which the economy of scale can be maintained.

In consequence, the product variety has increased greatly in the past few decades. The increase of product variety has taken place in almost every aspect of our lives, from radio broadcast stations to milk types, from KFC menu items to running shoe styles, etc (1998 Annual Report of the Federal Reserve Bank of Dallas). Take automobile industry as a specific example. The number of distinct vehicle models in the US increased from 44 in 1969 to 165 in 2005 (Ward’s Automotive Yearbook, 1970 & 2006). However, the increase of product variety also brings a lot of challenges to manufacturers and several studies have shown that high product variety has negative impact on manufacturing system performance, such as increasing manufacturing complexity, degrading quality, lowering productivity (MacDuffie et al., 1996; Fisher and Ittner, 1999).

Enabled by modular product designs, today’s manufacturers move from traditional non-modular assembly supply chains to modular assembly supply chains in order to mitigate the impact of product variety, as shown in Figure 4.1. Based on the modular product structure, the manufacturer in a modular assembly supply chain 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 and assembled by its suppliers and the manufacturer only does the final assembly of a few pre-assembled sub-assemblies. Modular assembly supply chains have been used widely in many industries, including automobile,
Figure 4.1: Manufacturers move from non-modular assembly supply chains to modular assembly supply chains

aerospace, electronics, etc. 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 (Fredriksson, 2006).

When the design structure of a modular product is decided, the manufacturer then needs to make the decisions about the manufacturing system layout and supply chain configuration. Usually based on a given modular product structure, there are more than one way to configure the supply chains, which will result in different number of echelons, intermediate sub-assemblers, etc. Most times the supply chain configuration decision is made through comparing the cost of different configurations. However, the presence of product variety brings many challenges to the cost analysis of modular assembly supply chains and in turn the decision of supply chain configuration. It is because sophisticated cost models for modular assembly supply chains with product variety are difficult to develop and analyze due to the network structure of an assembly supply chain and the influence of multiple products. In addition, cost models require the estimation of many parameters, e.g., manufacturing costs, overage and shortage costs, logistics costs, production and transportation leadtimes, some of which are not available in practice. Therefore, most research is based on empirical case studies instead of model-based analysis when investigating the effect of product variety on supply chain configurations (Randall and Ulrich, 2001; Salvador et al., 2004).

Due to these theoretical and practical challenges in using cost analysis, a new complexity measure for assembly supply chains has been proposed recently by Wang et al. (2009). This complexity measure is based on Shannon’s information entropy and does not require the estimation of supply chain parameters. It 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. Wang et al. (2009) also investigated the degree of consistency between cost and complexity of an assembly supply chain when comparing different supply chain configurations with the same level of product variety. Wang et al. (2009) found out that under certain conditions, 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 4.2, we review the relevant literature. In Section 4.3, we derive the complexity model based on Shannon’s information entropy by incorporating detailed information on supply chain structure and product variety. In Section 4.4, we theoretically investigate the supply chain configuration selection problem in two special scenarios. In Section 4.5, 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 4.6 gives some further discussion about this complexity research and Section 4.7 concludes the paper.

4.2 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 (1985) investigated a finite-horizon model
of a centralized assembly supply chain with two components. Rosling (1989) 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. Whang and Gerchak (2003) 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 (2004) 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 (2001) 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. (2004) 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 (2005) defined a complexity measure particularly applicable in product design; Cover and Thomas (1991) 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 (1948) 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,

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

(4.1)

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 (Wang et al., 1998; Chatzisavvas et al., 2005). For example, Deshmukh et al. (1998) 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. (2008) 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. (2002) developed an experimental methodology to study the operational complexity in a supplier-customer system. Following the similar method, Frizelle (2004) developed a metric to measure the complexity within the supply chain, called operational complexity. Wang et al. (2009) 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. (2008) 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. (2009) 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.
4.3 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 4.2. 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 \( m + 1, \ldots, n - 1 \) 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, Wang et al. (2009) define the following notations, which are also used in this paper:

\[ V_i : \text{the number of variants that node } i \text{ can produce, } i = 1, \ldots, n. \]

\[ S_i : \text{the set of nodes that are suppliers to node } i, i = 1, \ldots, n. \]
Figure 4.2: A general assembly supply chain and relationships of variants’ demand share

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

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 4.2 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., \(j \in S_i\)):

\[q_{ju} = \sum_{v \in A_{iju}} q_{iv}, j \in S_i. \quad (4.2)\]

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 4.2, 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, $\Phi$:

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

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 $j = 0, \ldots, n$. 

107
Step 2: For every supply-assembly relationship, where $\phi_{ji} = 1$ in matrix $\Phi$, matrix

$$Q_{ji} = ((q_{ij}^{uv})) = \begin{bmatrix} q_{i0}^{j} & \cdots & q_{iV_i}^{j} \\ \vdots & \ddots & \vdots \\ q_{iV_j}^{j} & \cdots & p_{jV_i}^{j} \end{bmatrix}_{V_j \times V_i}$$

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, $q_{iu}^{j}$, $i = 1, \ldots, V_j$, $v = 1, \ldots, V_i$ 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 $Q_{ji} = ((q_{ij}^{uv}))$ is normalized by

$$\tilde{q}_{uv}^{ji} = \frac{q_{uv}^{ji}}{K} \text{ where } K = \sum_i \sum_j \sum_u \sum_v q_{uv}^{ji} \quad (4.3)$$

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 (4.1), we define the complexity contribution of any supply-assembly relationship in the supply chain ($\phi_{ji} = 1$ in matrix $\Phi$), as follows:

$$C_{ji} = -\sum_u \sum_v \tilde{q}_{uv}^{ji} \log_2 \tilde{q}_{uv}^{ji} \quad (4.4)$$

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:

$$C = \sum_i \sum_j C_{ji} \quad (4.5)$$

The complexity of an assembly supply chain defined by Equation (4.5) can be calculated by a simpler formulation (4.6), 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,

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

(4.6)

Wang et al. (2009) also interprets this complexity measure through the following random experiment. Suppose for each node, a pool is formed by collecting the variants produced by that node, 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. (2009) showed that the information entropy of this random experiment is given by Equation (4.6). Loosely speaking, this 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 4.3. 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 \( \mathbf{q}_5 = \left( \frac{1}{7}, \frac{1}{8}, 0, \frac{1}{4}, 0, 0, \frac{1}{5}, 0 \right) \). Based on the demand vector of the final assembler, the demand vector of other nodes in the supply chain can be obtained by Equation (4.2) 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 \( \mathbf{Q}^{ji} \) is developed. For example, we have the following matrix \( \mathbf{Q}^{45} \) for \( \phi_{45} = 1 \),

\[
\mathbf{Q}^{45} = \begin{pmatrix}
\frac{1}{7} & \frac{1}{8} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{5} \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

Similarly, we get \( \mathbf{Q}^{01}, \mathbf{Q}^{02}, \mathbf{Q}^{03}, \mathbf{Q}^{14}, \mathbf{Q}^{24}, \mathbf{Q}^{35} \). Thirdly, matrix \( \mathbf{Q}^{ji} \) is normalized by Equation (4.3), where \( K = 7 \) in this example. For instance, we normalize \( \mathbf{Q}^{45} \) and obtain the following normalized matrix \( \tilde{\mathbf{Q}}^{45} \),

\[
\tilde{\mathbf{Q}}^{45} = \begin{pmatrix}
\frac{1}{74} & \frac{1}{56} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{1}{28} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{56} \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

Following the same procedure, we get other normalized matrixes, \( \tilde{\mathbf{Q}}^{01}, \tilde{\mathbf{Q}}^{02}, \tilde{\mathbf{Q}}^{03}, \tilde{\mathbf{Q}}^{14}, \tilde{\mathbf{Q}}^{24}, \tilde{\mathbf{Q}}^{35} \).
Fourthly, the complexity contribution of each supply-assembly relationship, based on each normalized matrix $\tilde{Q}^i_j$, is calculated through Equation (4.4). 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, $C = C_{01} + C_{02} + C_{03} + C_{14} + C_{24} + C_{35} + C_{45} = 4.029$.

### 4.4 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 4.1** 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 4.2 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 4.1 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 (4.2), 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 4.2** 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 4.2 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 4.3** 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, \( L_{m+i} = d + e_i, e_i \in \mathbb{Z}, \sum_{i=1}^{c} e_i = 0, i = 1, \ldots, c \). 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 4.3 provides some detailed insights regarding to 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 4.3 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 4.3, we can easily compare the complexity of the three configurations of Figure 4.4 and obtain the following complexity relationship, if the demand shares are the same for all the variants offered by the final assembler, $\text{Complexity}(I) > \text{Complexity}(II) > \text{Complexity}(III)$.

### 4.5 Optimal Assembly Supply Chain Selection

In section 4.4, 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
Figure 4.5: The example used to illustrate the methodology of finding the optimal assembly supply chain

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.

4.5.1 Optimal Assembly Supply Chain Selection without Assembly Constraints

For easy explanation, a specific example shown in Figure 4.5, 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 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 4.5. 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 4.3). For example, here we name the final assembler in Figure 4.5 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 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 (4.2) and substitute them back to Equation (4.6) 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 4.6. For each configuration, there are several possible supply chain candidates. For example, the configuration IV in Figure 4.6 has three different supply chains due to the location difference of the nodes in the most upstream echelon, as shown in Figure 4.7.

In order to enumerate all possible supply chain candidates, considering the 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 (2005) 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 4.7 (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 4.7 (a).

After developing this description method for assembly supply chains, we then develop an iterative decomposition algorithm, that generates all possible assembly supply chain

![Figure 4.7: There are more than one supply chain candidates within one configuration](image-url)
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 4.5, 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 4.5, 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) 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 4.8. 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
Figure 4.8: Iterative decomposition algorithm to generate all possible supply chain candidates and × stands for the infeasible candidates, which will be discussed in section 4.5.2

Equation (4.2) and then the corresponding complexity of that assembly supply chain is calculated through Equation (4.6). The optimal assembly supply chain is achieved through comparing complexity of all supply chain candidates and selecting the one with the least complexity.

### 4.5.2 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 4.5, 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 4.9: 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:

\[
A \succ C, B \succ C, C \succ D
\]

(4.7)

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 4.5.1, 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 4.8, where \( \times \) stands for the infeasible supply chain candidates that do not satisfy constraints (4.7).

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 4.5.1, we keep decomposing the assembly combinations obtained in step 2 until the cardinality of each inner parenthesis

\[
\text{Figure 4.9: A set of assembly sequence constraints}
\]
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 4.10. Compared with the previous method, the number of feasibility check is reduced, because infeasible intermediate supply chain candidates stop further decomposition and no more feasibility check is performed from this point.

Here the example shown in Figure 4.5, 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 (4.7). 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 4.10 summaries the details of this method. This method helps to reduce the number of feasibility check from 26 to 13.

In the iterative decomposition algorithm developed in section 4.5.1, 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 4.11 summaries the details of this feasibility check criteria.

We use Figure 4.10 as an example to illustrate this procedure. In first iteration, \( P^1 = (ABCD) \) is decomposed into 14 intermediate assembly combinations. We choose one intermediate combination, \(((ABC)(D))\), to check whether it satisfies constraints (4.7). 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
Figure 4.11: Feasibility check criteria for an intermediate assembly supply chain

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.

4.6 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 4.3, 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 4.8, 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 (4.6), 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.

4.7 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.
Appendix: Proofs of Propositions

Proof of Proposition 4.1

Proof of (a)

The demand vector of the final assembler, node $n$, is $q_n := (q_{n1}, q_{n2}, \ldots, q_{nV_n})$. 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., $q_{n1} \to 1$, the demand share of variant 1 at node $i = 1, 2, \ldots, n-1$ also increases and approaches to 1, i.e., $q_{i1} \to 1$. Because $\sum_{j=1}^{V_i} q_{ij} = 1$ and $q_{i1} \to 1$ for $i = 1, \ldots, n$, then the demand share of other variants at node $i$ must approach 0, i.e., $q_{ij} \to 0, j = 2, \ldots, V_i$.

The complexity of an assembly supply chain defined in Equation (4.6) is $C = -\sum_{i=1}^{n} L_i \sum_{v=1}^{V_i} \frac{q_{iv}}{K} \log_2 \frac{q_{iv}}{K}$, 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 (4.6) can also be written as

$$C = \log_2 K - \frac{1}{K} \sum_{i=1}^{n} L_i \sum_{v=1}^{V_i} q_{iv} \log_2 q_{iv}. \quad (4.8)$$

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

$$C = \log_2 K - \frac{1}{K} \sum_{i=1}^{n} L_i (q_{i1} \log_2 q_{i1} + \sum_{v=2}^{V_i} q_{iv} \log_2 q_{iv}). \quad (4.9)$$

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

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

\[
= \lim_{q_{iv} \to 0} -\frac{1}{\ln 2 \cdot q_{iv}} = \lim_{q_{iv} \to 0} \frac{-q_{iv}}{\ln 2} = 0.
\]

Then in the scenario of one dominant variant, the complexity of an assembly chain, represented as Equation (4.9), 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,

\[
K = (n - 1) + m \tag{4.10}
\]

Since \( m \) is fixed by the number of module in the final product, in order to obtain the minimum value of \( K \) in Equation (4.10), 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 4.2**

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 (4.2), it is easily to verify that the demand shares are also identical across all the variants at other nodes in the supply chain.
Figure 4.12: One non-modular assembly supply chain and one modular assembly, both of which have one intermediate sub-assembler

In addition, here we assume all the nodes in the most upstream echelon, node 1,...,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 \( \mathbf{q}_i = (\frac{1}{V}, \ldots, \frac{1}{V})_{1 \times V}, i = 1, \ldots m \). 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, \( \mathbf{q}_n = (\frac{1}{V_n}, \ldots, \frac{1}{V_n})_{1 \times V_n} = (\frac{1}{V}, \ldots, \frac{1}{V})_{1 \times V^m} \). One is non-modular assembly supply chain in Figure 4.12(a) and another is modular assembly supply chain in Figure 4.12(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 \( \mathbf{q}_{n-1} = (\frac{1}{V}, \ldots, \frac{1}{V})_{1 \times V^{m-1}} \). The complexity of an assembly supply chain can be calculated through Equation (4.8), 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 \( \mathbf{q}_i = (\frac{1}{V}, \ldots, \frac{1}{V})_{1 \times V}, L_i = 1, i = 1, \ldots m, \mathbf{q}_n = (\frac{1}{V}, \ldots, \frac{1}{V})_{1 \times V^m}, L_n = m \) and \( K = 2m \) back to Equation (4.8) and get the
complexity of the non-modular assembly supply chain in Figure 4.12(a),

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

Following the same argument, by substituting \( q_i = (\frac{1}{V_{i-1}}, \ldots, \frac{1}{V})_{1 \times V}, L_i = 1, i = 1, \ldots m, \)
\( q_{n-1} = (\frac{1}{V_{m-1}}, \ldots, \frac{1}{V})_{1 \times V^{m-1}}, L_{n-1} = m - 1, q_n = (\frac{1}{V^{m}}, \ldots, \frac{1}{V})_{1 \times V^m}, L_n = 2 \) and
\( K = 2m + 1 \) back to Equation (4.8), we can get the complexity of the modular assembly supply chain in Figure 4.12(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 4.12 is,

\[ C_{\text{mod}} - C_{\text{non-mod}} = \log_2 \frac{2m + 1}{2m} - \frac{m - 1}{2(2m + 1)} \log_2 V \quad (4.11) \]

Since \( m \) is the number of nodes in the most upstream echelon of the assembly supply chain, \( m \) must a integer greater than 1, i.e., \( m > 1, a \in \mathbb{Z} \), 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 (4.11), 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 4.12(b), has less complexity value than the corresponding non-modular assembly supply chain, shown in Figure 4.12(a).

**Proof of Proposition 4.3**

Suppose the demand shares are equal across all the variants provided by the final assembler, node \( n \), i.e., \( q_n = (\frac{1}{V_{n-1}}, \ldots, \frac{1}{V_0})_{1 \times V_n} \). By Equation (4.2), 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., $m = c \times d, c, d \in \mathbb{Z}^+$. 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 $q_i^{(1)} = (\frac{1}{V_d}, \ldots, \frac{1}{V_d})_{1 \times V_d}, i = m + 1, \ldots m + c$. Supply chain 2 has $d$ intermediate sub-assemblers, each of which has $c$ suppliers from the most upstream echelon and then has $q_i^{(2)} = (\frac{1}{V_c}, \ldots, \frac{1}{V_c})_{1 \times V_c}, i = m + 1, \ldots m + d$.

Recall in the proof of Proposition 4.1, the complexity of an assembly supply chain can be calculated through Equation (4.8), 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 supply chain 1, there are total $2m + c$ arcs in the supply chain, i.e., $K^{(1)} = 2m + c$. We substitute $q_i^{(1)} = (\frac{1}{V}, \ldots, \frac{1}{V})_{1 \times V}, L_i^{(1)} = 1, i = 1, \ldots m$ and $q_i^{(1)} = (\frac{1}{V_d}, \ldots, \frac{1}{V_d})_{1 \times V_d}, L_i^{(1)} = d, i = m + 1, \ldots m + c$ and $q_n^{(1)} = (\frac{1}{V_m}, \ldots, \frac{1}{V_m})_{1 \times V_m}, L_n^{(1)} = c$ and $K^{(1)} = 2m + c$ back to Equation (4.8) 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 $q_i^{(2)} = (\frac{1}{V}, \ldots, \frac{1}{V})_{1 \times V}, L_i^{(2)} = 1, i = 1, \ldots m$ and $q_i^{(2)} = (\frac{1}{V_c}, \ldots, \frac{1}{V_c})_{1 \times V_c}, L_i^{(2)} = c, i = m + 1, \ldots m + d$ and $q_n^{(2)} = (\frac{1}{V_m}, \ldots, \frac{1}{V_m})_{1 \times V_m}, L_n^{(2)} = d$ and $K^{(2)} = 2m + d$ back to Equation (4.8) 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 $A = \log_2 \frac{2m+c}{2m+c}$ 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., $L_{m+i}^{(1)} = d$, $i = 1, \ldots, c$, and $\sum_{i=1}^{c} L_{m+i}^{(1)} = cd = m$. 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 the these $c$ sub-assemblers is still $cd$, i.e., $\sum_{i=1}^{c} L_{m+i}^{(2)} = cd = m$. We can rewrite the number of suppliers for these $c$ sub-assemblers in supply chain 2 in the following form, $L_{m+i}^{(2)} = d + e_i, e_i \in Z, \sum_{i=1}^{c} e_i = 0, i = 1, \ldots, c$.

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

The complexity of an assembly supply chain can be calculated through Equation (4.8), i.e., $C = \log_2 K - \frac{1}{K} \sum_{i=1}^{n} L_i \sum_{v=1}^{V_i} q_{iv} \log_2 g_{iv}$. We substitute $q_i^{(1)} = (\frac{1}{V^{d}}, \ldots, \frac{1}{V^{n}})_{1 \times V}, L_{i}^{(1)} = 1, i = 1, \ldots, m$ and $q_{m+i}^{(1)} = (\frac{1}{V^{d}}, \ldots, \frac{1}{V^{n}})_{1 \times V^{d}}, L_{m+i}^{(1)} = d, i = 1, \ldots, c$ and $q_i^{(1)} = (\frac{1}{V^{d}}, \ldots, \frac{1}{V^{n}})_{1 \times V^{d}}, L_i^{(1)} = c$ and $K^{(1)} = 2m + c$ back to Equation (4.8) 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 $q_i^{(2)} = (\frac{1}{V^{d}}, \ldots, \frac{1}{V^{n}})_{1 \times V}, L_i^{(2)} = 1, i = 1, \ldots, m$ and $q_{m+i}^{(2)} = (\frac{1}{V^{d+e_i}}, \ldots, \frac{1}{V^{n+e_i}})_{1 \times V^{d+e_i}}, L_{m+i}^{(2)} = d + e_i, i = 1, \ldots, c$ and $q_i^{(2)} = (\frac{1}{V^{d}}, \ldots, \frac{1}{V^{n}})_{1 \times V^{d}}$, $L_i^{(2)} = c$ and $K^{(2)} = 2m + c$ back to Equation (4.8) to get the complexity of the supply chain 2,
c and $K^{(2)} = 2m + c$ back to Equation (4.8) 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 $V \geq 1$, $m, c \in Z^+$ 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.
BIBLIOGRAPHY


CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusions and Original Contributions

This dissertation presents original research work in modeling the complexity induced by product variety in mixed-model assembly systems and supply chains and investigates the effect of product variety on the performance of assembly systems and supply chains. The major achievements of this dissertation are summarized in the following five parts.

1. Definition of a complexity measure for mixed-model assembly systems with different configurations

A complexity measure is proposed for mixed-model assembly systems with different configurations, including serial, parallel and hybrid of the two. The complexity measure is based on the concept of information entropy and takes into account operator choices at each station and system configurations.

2. Investigation of the impact of product variety induced complexity on the performance of mixed-model assembly systems

An approximate throughput model is developed for mixed-model assembly systems by taking the complexity-based operator reaction time and fatigue effect into consideration. Then the complexity and throughput models are used to compare the performances of assembly systems with different configurations. It is discovered that the complexity increases as the configuration changes from serial to hybrid and then
to parallel. The throughput decreases as fatigue effect increases from none to low and to high. For the same level increase of fatigue, the throughput of mixed-model assembly systems with higher complexity reduces more than the throughput of those with lower complexity.

3. **Methods for differentiating variants at parallel stations to reduce the complexity and increase the throughput of a mixed-model assembly system**

A mathematical formulation is developed based on mixed-binary nonlinear programming to minimize the complexity or maximize the throughput of a mixed-model assembly system through allocating the modules to different stations and differentiating the production of variants at parallel stations.

4. **Definition of a complexity measure for assembly supply chains**

A complexity measure is proposed for assembly supply chains, based on the concept of information entropy. The complexity measure takes into account factors such as the supply chain configuration, the level of variety offered by each node of the supply chain, and the demand mix ratios across all the variants offered by a node of the supply chain.

5. **Investigation of the relationship between the complexity and cost of an assembly supply chain**

The relationship between the complexity and cost of an assembly supply chain is investigated. The degree of consistency between the complexity and cost is studied when they are used to compare assembly supply chains with the same configuration, but different levels of product variety. It is shown that the complexity and cost are equivalent under certain conditions, in the sense that both of them rank a given set of supply chains in the same order. Even when these conditions do not hold, a numerical study demonstrates that the complexity and cost criteria rank supply chains consistently in an overwhelming majority of cases. In addition, the agreement between the cost and complexity criteria is studied when comparing assembly supply chains with the same level of product variety, but different configurations. In such
cases, the results show that the consistency between cost and complexity criteria is not very reliable.

6. Application of supply chain complexity to configuration design

The complexity measure for assembly supply chains is applied to configuration design, i.e., determining the optimal 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 studied in the following two special scenarios: 1) there is only one dominant variant among all the variants offered by the final assembler, i.e., the demand share of one particular variant is bigger than the demand share of others at the final assembler, and 2) demand shares are equal across all variants at the final assembler. It is shown that in the first scenario, the optimal assembly supply chain should be non-modular; but in the scenario of equal demand shares, modular supply chains are more beneficial than non-modular ones when the product variety is high. For the scenario of general demands, a methodology is developed to find the optimal supply chain with/without assembly sequence constraints.

The original contributions of this dissertation are summarized as follows:

1. A complexity measure is proposed for mixed-model assembly systems, which takes into account the factors of operator choices at each station and system configurations, including serial, parallel and hybrid. It extends complexity research from pure serial assembly lines to assembly systems with different configurations.

2. An approximate throughput model is developed for mixed-model assembly systems by taking operator reaction time and fatigue effect into consideration. It helps us better understand how product variety impacts the performance of the human operator at each station and in turn influences the performance of mixed-model assembly systems.

3. The complexity and throughput models are used to compare the performances of assembly systems with different configurations. The developed methodology and analysis results can be used as tools for system designers in evaluating mixed-model assembly systems with different configurations.
4. A new performance measure is developed for assembly supply chains, complexity. This complexity measure takes into account the following factors, the supply chain configuration, the level of variety offered by each node of the supply chain, and the demand mix ratios across all the variants offered by a node of the supply chain. In addition, the complexity measure does not require the estimation of cost parameters of supply chains.

5. The relationship between the complexity and cost of an assembly supply chain is studied in the following two scenarios: comparing assembly supply chains with the same configuration but different levels of product variety and comparing supply chains with the same levels of product variety but different configurations. The study provides the practical guidelines on when and how to use the complexity measure of assembly supply chains.

6. The complexity model of assembly supply chains is applied to the configuration design, i.e., how to find the optimal supply chain configuration given the number of variants offered and their demand mix ratios. The developed methodologies can be used as model-based analysis tools, instead of empirical studies, to study the configuration design problem for assembly supply chains.

5.2 Future Research

The methodologies and models developed in this research could be further improved and extended in the following directions:

1. To compare mixed-model assembly system configurations in terms of complexity and throughput:

   Current configuration comparison of mixed-model assembly systems is based on all symmetric configurations generated by the distribution algorithm developed in Web-bink and Hu (2005). As the number of stations in the system increases, the number of generated configurations increases exponentially, which makes the configuration comparison difficult to conduct. Theoretical performance analysis of different con-
figurations can help reduce the number of configurations that need to be compared. For example, parallel-serial configurations with crossover always have higher throughput than the same parallel-serial configurations without crossover as shown in Figure 5.1 (Freiheit et al., 2004). Therefore, when we compare the throughput of different configurations, only parallel-serial configurations with crossover need to be included.

2. To develop heuristic algorithms to find the mixed-model assembly system with minimum complexity and the one with maximum throughput:

The mixed-binary nonlinear programming (MBNP) is used in this dissertation to find the assembly system with minimum complexity and the system with maximum throughput. However, solving MBNP problems could be challenging when the size of the problem is big. The solution is highly dependent on the solver, the start point and the search algorithm, and sometimes, the solution found is a local optimum instead of the global optimum. Developing heuristic algorithms can provide another alternative solution. The future research can focus on how to develop highly efficient algorithms that can find the optimal assembly system quickly and accurately.

3. To study the relationship between complexity and other performance measures of assembly supply chains:

In this dissertation, we study the relationship between the complexity and cost of an assembly supply chain, and our study suggests that complexity is a fairly good performance measure for assembly supply chains, especially in comparing supply chains with the same configuration but different levels of product variety. One would expect that studying the relationship of complexity and other performance measures
of supply chains will make this complexity measure more attractive and useful. For example, the lead time and the robustness to the disruption are two important performance measures for supply chains, which can be used to validate the complexity measure (de Treville et al., 2004; Klibia et al., 2010). Future research will explore more application areas for the complexity measure of assembly supply chains.

4. To improve the efficiency of solving the configuration design problem for assembly supply chains:

The complexity measure of assembly supply chains has been applied to the configuration design in this dissertation, i.e., to find the assembly supply chain with minimum complexity for a given the product variety level. One challenge of this application is that as the number of product modules increases, the number of supply chain candidates generated by the iterative decomposition algorithm increases exponentially, which makes the comparison difficult to conduct. Reformulating the problem by some mathematical programming tools, such as mixed integer programming (MIP) and dynamic programming (DP), could provide another alternative solution. Therefore, it will be an interesting research topic to study how to formulate the configuration design of assembly supply chains as an optimization problem in a more efficient way so that it can be solved quickly and accurately, even if the size of the problem is big.
BIBLIOGRAPHY


