Concurrent Design of Assembly Plans and Supply Chains: Models, Algorithms, and Strategies

by

Heng Kuang

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mechanical Engineering)
in The University of Michigan
2016

Doctoral Committee:

Professor S. Jack Hu, Co-chair
Professor Jeonghan Ko, Co-chair
Professor Xiuli Chao
Professor Panos Y. Papalambros

DEDICATION

This dissertation is dedicated to my family: my mother, father, and brother. Thank you for all of your love and support throughout my life.

ACKNOWLEDGEMENTS

I would like to express my special appreciation and thanks to my advisors Professor S. Jack Hu and Professor Jeonghan Ko for all their continuous support and advice. It was always a pleasant experience to meet with Prof. Hu. His curiosity, humility, patience, and trust in students were contagious and motivational. What I learned from him during the past four and half years will be my greatest asset. Prof. Ko was both a mentor and a friend to me. His advice and help in research, career development, and work-life balance has always been beneficial.

I would like to thank Professor Xiuli Chao and Professor Panos Y. Papalambros for serving as my committee members. Prof. Chao's expertise and insights in operation research were inspiring. It was in his IOE 512 that I came up with the idea to model my research problem using dynamic programming. I want to thank Prof. Papalambros for providing me a good opportunity to participate in the ARC modularity project, which opened another window for me to observe how top researchers conducted research.

I gratefully acknowledge the financial support from the U.S. National Science Foundation (NSF) grants (Nos. 1068029 and 1331633).

I also want to thank Professor Jing Sun, who encouraged me to apply to the University of Michigan (UM), when she was visiting University of Science and Technology of China. She has been supporting and motivating me since I came here in

2010. My appreciation also goes to Professor Semyon M. Meerkov, who taught me a lot in my first year at UM.

My gratitude also goes to my colleagues and friends in Hu Lab with whom I had the privilege to work and spend time together. They include but are not limited to: Professor Mihaela Banu, Professor Hui Wang, Professor Changbai Tan, Professor Haseung Chung, Professor Jingjing Li, Sam, John, Chenhui, Grace, Liang, Kevin, Daniel, and Ying. My time at UM was made enjoyable in large part due to the many friends that became a part of my life. Thank you.

I am deeply thankful to my family for their love, support, and sacrifices. Especially I want to thank my older brother, who has been in company with my parents in these years, which allowed me to fully focus on my Ph.D. study.

Finally, I want to thank the University of Michigan, in which place I met all these nice people and had the wonderful time.

Go blue.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

ABSTRACT

Concurrent Design of Assembly Plans and Supply Chains: Models, Algorithms, and Strategies

By

Heng Kuang

Co-Chairs: S. Jack Hu and Jeonghan Ko

Assembly planning and supply chain designs are two inter-dependent activities in product development. The traditional sequential approach of designing the supply chain after completing assembly planning results in long lead time for product realization and sub-optimal product cost. The weakness of the sequential method is exacerbated nowadays as product proliferation brings more challenges to assembly system design and supply chain management. Making concurrent decisions on assembly plans and supply chain configurations is a desirable strategy. However, due to the complexity of both assembly representations and supply chain modeling, there have been limited systematic models,

optimization algorithms, or deep understanding of the interaction between assembly-plan and supply-chain designs.

This dissertation first analyzes and compares existing assembly representation methods. Hyper AND/OR Graph (HAG) is then developed to incorporate both assembly planning and supply chain configuration information by adding one additional layer representing supplier information on top of a typical assembly AND/OR graph. Based on HAG, a DP based algorithm with a polynomial complexity for typical assembly products is developed to generate the assembly plans and supplier assignment at the optimal cost. For the problem with a lead time constraint, a revised DP algorithm with a pseudo-polynomial complexity is also presented. Under the scenario of product family designs, an investigation is carried out on the optimal strategies to design assembly supply chains when commonality is limited between products in the family. The impact of product variety on safety inventory is derived and then evaluated with a performance measure. Strategies of prioritized differentiation and branch balancing are suggested for optimal process sequencing and assembly decomposition.

The outcome of this research are threefold: (1) it establishes a foundation for the research on integrated designs of assembly plans and supply chains as well as other concurrent design problems; (2) it offers a tool for integrated assembly plan and supply chain designs using which manufacturers can shorten the product development time, lower the product cost, and increase the responsiveness to fluctuations in supply chains; and (3) it provides a measure of the impact of product variety on inventory and insightful strategies to manage complicated assembly supply chains.

CHAPTER 1

Introduction

1.1 Motivation

Manufacturers worldwide are making efforts to increase their competitive capabilities in response to the global markets, diverse customer demands, and rapidly changing supplier environment. Competitive advantages can be gained by considering manufacturing issues in the three phases of product planning: product design, assembly planning, and supply chain configuration. Traditionally, decisions on these three phases are made in a serial pattern. First, one or several product designs are selected from a set of feasible designs considering market objectives and engineering constraints. Second, feasible assembly plans are developed, including decomposing the final product into subassemblies and generating assembly sequences. The assembly plan is guided primarily by operational objectives and manufacturing capabilities. Finally, the supply chain is configured under the constraints of the product and assembly designs. This serial pattern is known to generate solutions that suffer from two major deficiencies [1]. The first deficiency is the long lead time for product realization because there are often iterative changes among the different stages of decision makings. The second deficiency is sub-optimality because it is usually difficult to assess the designs in the early two stages since costs in these stages are difficult to define and quantify. The deficiencies are exacerbated by the product variety and demand uncertainty as assembly plans and supply chains become more difficult to manage in a product family scenario.

To overcome the shortcomings of sequential uncoordinated designs, Concurrent Engineering (CE) has received attention from both industry and academia. Over the last three decades, various aspects of CE have been studied, ranging from combining production considerations with product designs to incorporating supply chain issues with product design and assembly planning. For example, a survey of the U.S. auto makers reveals that suppliers have become more active in participating in subassembly and manufacturing plan designs [2]. CE reduces re-design and rework, and leads to smoother product launch. CE applications were reported to achieve a 30-60% reduction in time-to-market, 15-50% reduction in product life-cycle costs, and 55-95% reduction in engineering changes and rework [3]. However, it complicates the design problem because it requires joint optimization with larger constraint and variable sets.

Concurrent design requires sophisticated coordination between assemblers and suppliers from manufacturing to supply. For a complex assembly product, a small change in a subassembly will likely propagate through the assembly hierarchy, leading to numerous changes in component design, re-tooling in manufacturing, and logistics and contractual changes in supply chains. The particular area worth significant attention is how to adapt product modules for assembly and supply chains in a single product as well as product family scenarios. Literature review shows that there lacks systematic research on coordinating assembly decomposition in manufacturing and supply chains. Therefore, the following four questions are considered in this research:

1. How to represent assembly constraints and supply chain configurations cohesively?
2. How to coordinate the assembly processes and supply chains given the manufacturing and supply resources?

3. How to apply the concurrent assembly plan and supply chain design method to industrial products and guide the designs of assembly plans and supply chains?

4. How to integrate the designs of assembly plans and supply chains for a product family so as to manage the variety induced complexity?

1.2 Summary of Literature Review

A summary of the review is provided below, including assembly and supply chain representation, concurrent design, optimization algorithms, and management of complexity incurred by variety.

Assembly planning problems are usually modeled as graph decomposition or sequence generation problems, while supply chain problems are usually represented as network optimization ones. Due to the different characteristics of the two types of decision making problems, there has been no widely accepted model to represent the integrated decisions of assembly plan and supply chain.

While various manufacturing factors such as tooling and processes, e.g., machining and additive manufacturing, have been considered in the concurrent design of manufacturing systems and supply chain, few studies investigated the relationship between the assembly plan and the supply chain, i.e., how assembly plans can affect supply chains or how assembly plans should be designed given supply chain configurations.

Only a few researchers have conducted research on integrated assembly planning and supply chain designs. However, due to the lack of integrated modeling methods for assembly plan and supply chain, the decision space is huge, even for products with small complexity. Approximate optimization algorithms, such as Genetic Algorithm (GA), were adopted in those researches to solve the integrated optimization problem. Therefore, this dissertation seeks to develop a model to integrate assembly plan and supply chain decisions and hence to support systematic studies on the concurrent decision makings.

The research on the management of product variety induced complexity can be generally divided into two categories: index-based optimization and heuristic strategies. Although potential insights into efficient assembly supply chain designs were revealed in index-bases optimization research, there lacked explicit relations between the proposed indices and common measures such as costs, time, and quality. The heuristic strategy studies often assumed that common processes exist throughout product families. However, the commonality might exist only partially in today's manufacturing where variety of products is a key competitive factor. Thus, many important questions remain unexplored, such as optimal strategy when commonality is limited or more generally the room for further optimization in addition to delayed differentiation.

1.3 Research Objectives and Tasks

The objective of this dissertation is to develop models, algorithms, and strategies to coordinate the complicated designs of assembly plans and supply chains. After an initial attempt to formulate the optimization problem in the most generic form, the two barriers in the formulation will be outlined. Due to the complexity of assembly constraints, we will compare and analyze the different assembly representations and assess the possibility to combine them with supply chain information and their applicability to the optimization problem. A model will then be built based on the selected assembly representation to incorporate the supply chain configurations. Secondly an algorithm that fully exploits the structural merits of the established model will be developed to solve the integrated optimization problem. Finally, we will extend the discussion to product families, in which we will derive the optimal strategy to manage assembly supply chains in order to mitigate the variety induced complexity and risks. The proposed approach is summarized in six steps, as shown in Figure 1.1.

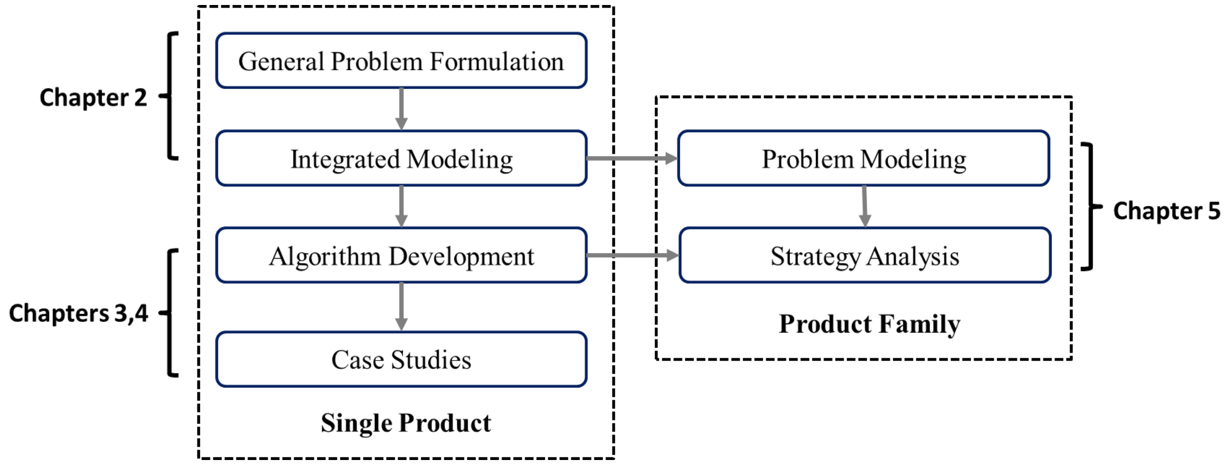


Figure 1.1. Flow chart of the proposed approach

The outcome of this research includes a mathematical model to support joint optimization for assembly plan and supply chain decisions for an individual product and optimal strategy to manage assembly supply chains in a product family setting. More specifically, the contributions include

- Development of a mathematical model to integrate assembly plan and supply chain decisions;
- A systematic analysis of the problem of integrated design of assembly plan and supply chain and the development of a computationally efficient algorithm to generate the optimal configurations;
- A method to assess assembly plans through the perspective of total supply chain cost and discussions on possible application of our findings.

- A tool to guide the integrated design of assembly plans and supply chains, enabling quick prototyping and reconfigurations.
- An optimal strategy to manage assembly supply chains through mitigating the variety induced risks for a product family.

By using this model, manufacturers could make assembly plan and supply chain decisions simultaneously, leading to a reduced number of design iterations, shortened lead times, and lower total cost. Moreover, manufacturers could design more robust assembly plans to offset the supply availability and cost variation in supply chain through better understandings of the relationship between assembly plan and supply chain. Last but not the least importantly, the research on assembly supply chain designs of a product family deepens the understanding of the impact of assembly plans on supply chains.

1.4 Organization of the Dissertation

The remainder of this dissertation is organized as follows. In Chapter 2, the integrated design problem is formulated as a general optimization problem, and the main research barriers are discussed. After a comprehensive analysis on the assembly representations, an enhanced AND/OR graph is developed to include both assembly plan and supply chain information. In Chapter 3, a DP algorithm is developed for coordinated optimization problems and case studies are conducted. Chapter 4 considers the coordination problem with a lead time constraint, which could be categorized as a resource constrained optimization problem. An algorithm is developed based on the one in Chapter 3 and its computational complexity is thoroughly analyzed. A case study is also presented in this chapter. Chapter 5 focuses on the modeling and analysis of the impact of assembly plans on supply chain management under a product family scenario. An optimal strategy is suggested. Chapter 6 concludes the dissertation and discusses the future work.

CHAPTER 2

Integrated Modeling of Assembly Plan and Supply Chain

2.1 Introduction

As discussed in the introduction, a question remains to be answered is in how to seamlessly integrate assembly plans and supply chains in one integrated model. This chapter starts with a literature review and identifies the research gaps. Then a simple example illustrates the concurrent design problem of assembly plans and supply chains. The concurrent design problem is formulated as an optimization problem with the goal of minimizing the total cost. After the identification of the challenges of the problem, existing methods for assembly representation will be reviewed and a model to incorporate both assembly and supply chain information will be developed.

In assembly modeling, graph-based methods are commonly used for assembly representation. They represent topological relationships between the parts of an assembly. Liaison Diagram was used by Bourjault to represent the product structure [4]. Various representations have been proposed to model assembly sequences. Homem De Mello and Sanderson applied an AND/OR graph to represent assembly plans [5]. The AND/OR Graph is a directed hyper-graph featured with AND and OR relationships. While AND represents a composition relationship between an assembly and its subassemblies, OR represents multiple, optional assembly methods of a product or a subassembly. Wolter developed State Diagram with connection states between components representing assembly states of a product and with transitions between the assembly states representing assembly work [6]. The relationship between the AND/OR graph and State Diagram, e.g., the transformations between two representations and model complexity comparison, was studied by Homem de Mello and Sanderson [7].

Significant research has been carried out in generating assembly sequences based on various representations. Bourjault [4] as well as De Fazio and Whitney [8] presented means for generating all assembly sequences algorithmically by asking questions about the mating of part pairs. Homem de Mello and Sanderson transformed the assembly sequence problem into an assembly decomposition problem and then represented the assembly sequences in an AND/OR graph [9]. Ong and Wong [10] developed an algorithm to automatically detect subassemblies using the combined information of liaison and precedence graphs. Knosla and Mattikali [11] developed a method to generate assembly sequences from a 3D model by detecting possible collisions. Gao, Xiang, and Duan [12] applied gray clustering to subassembly identification based on graph system theory. Lee [13] constructed a weighted abstract liaison graph (*WALG*) to extract subassemblies based on stability and structural connectivity associated with liaisons. Li et al. [14] developed a system supporting automatic generation of an assembly system configuration with equipment selection and optimal manufacturing cost. Ko et al. [15] studied assembly decomposition considering its impact on the final product quality. Fujimoto, Fuji, and Nagata [16] introduced a modified genetic algorithm (GA) to cope with sequence nonlinearity and constraints in assembly planning. Xu and Liang [17] applied a modified Chebyshev goal programming approach to solve the multi-objective problem of concurrent optimization of product module selection and assembly line configuration.

In the 1980s, Supply Chain Management (SCM) addressed the need in integrating the key business processes, from upstream suppliers to end customers [18]. Generally speaking, a supply chain is a network of nodes, which could be enterprises engaged in activities ranging from the supply of raw materials to the production and delivery of end-products to target customers. Each node in the supply chain network often has several alternative options for accomplishing its functions. Deciding which options should be used at each node and deciding where inventory should be placed is referred to as Supply Chain Configuration (SCC) [19]. Given assembly modules, supplier selection has been studied from various perspectives. A supply chain is often modeled as a multi-stage production and inventory network under a periodically reviewed base-stock policy. Graves and Willems [20] developed an SCC optimization model that minimizes the total supply chain cost including the safety stock cost, pipeline stock cost and cost of goods sold. Thoneemann and Bradley [21] investigated the impact of product variety on supply chain performance from several different perspectives. Viswanadham and Gaonkar [22] studied partner selection and synchronized planning in a dynamic manufacturing network. Torabi and Hassini [23] provided an efficient production plan that integrates the procurement and distribution plans into a unified framework. Gunasekaran, Lai, and Cheng [24] analyzed both Agile Management and Supply Chain Management with the objective of developing a framework for responsive supply chains. Williamson [25] examined outsourcing from the perspective of transaction costs. Fawcett, Magnan, and McCarter [26] provided a quantitative and qualitative analysis of the benefits, barriers, and bridges to successful collaboration in strategic supply chains. Manuj and Mentzer [27] explored the phenomenon of risk management and risk management strategies in global supply chains.

Assembly planning problems are usually modeled as graph decomposition or sequence generation problems while supply chain problems are typically modeled as network optimization ones. Due to the different characteristics of the two decision-making problems, there have been no widely accepted models to represent the integration of assembly plan and supply chain decisions. This chapter focuses on a model to integrate assembly planning and supply chain decisions.

2.2 Problem Illustration

Boujault's pen example is used to illustrate the combined decision making problem. The pen consists of six components: body (A), head (B), cartridge (C), ink (D), button (E), and cap (F), as shown in Figure 2.1. The components are joined by five connections or liaisons: 1 (body A to head B), 2 (button E to body A), 3 (head B to cartridge C), 4 (cartridge C to ink D), and 5 (cap F to body A), as represented by the graph in Figure 2.2. This type of graphs is called the liaison graph, in which nodes represent components and edges represent connections between the components. Realizing a liaison is equivalent to performing an assembly process or task.

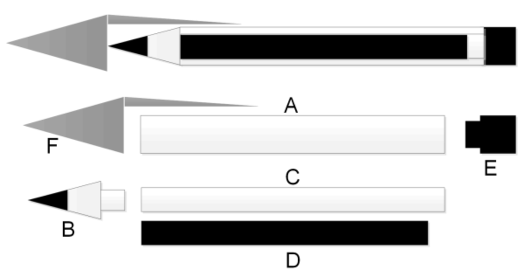


Figure 2.1. A pen example: assembly and its components, adapted from [28].

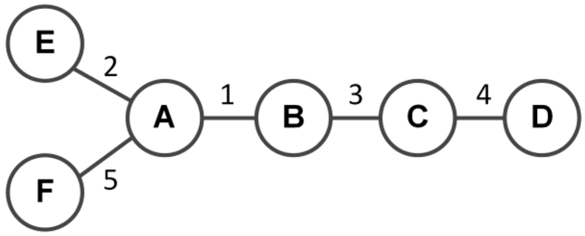


Figure 2.2. Liaison graph of the pen, adapted from [28].

For the supply chain of the pen assembly, assume that components A, B, C, and D can be purchased only in Area 1 while E and F only in Area 2. Assume further that the suppliers in Area 1 possess specialties only for assembly processes 1, 3, and 4, while assembly processes 2 and 5 can be performed only by the suppliers in Area 2. If Areas 1 and 2 are far away from each other, the transportation cost between them can be one of the most significant cost elements.

There are various ways to assemble the pen, among which the following two plans (sequences) are chosen for illustration:

Plan I: 2→4→3→1→5,

Plan II: 1→4→3→5→2.

If Plan I is selected, the assembly plan and transportation flow are as follows:

Purchase A in Area 1 and transport A to Area 2;

Purchase E in Area 2, finish task 2 to assemble A and E together, and then transport AE back to Area 1;

Purchase B, C, and D, and finish tasks 4, 3, and 1 to build subassembly ABCDE in Area 1, and then transport ABCDE to Area 2;

Purchase F and then assemble F with ABCDE for the final product.

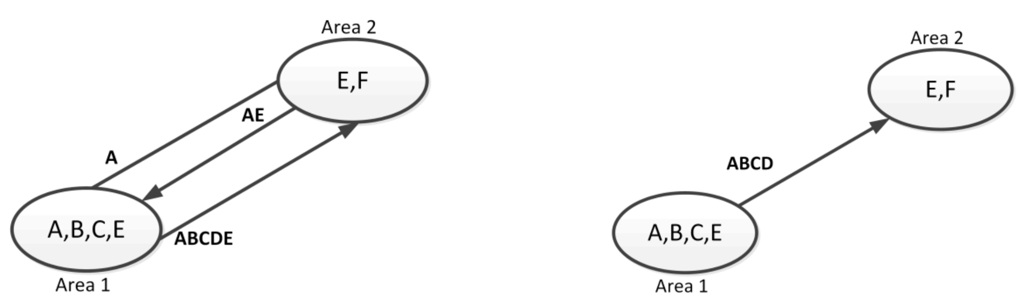
As shown in Fig. 3.3(a), there are repeated transportation flows between two areas due to the inappropriate assembly plans.

Instead, if Plan II is adopted, the transportation flow could be simplified:

A, B, C, and D will be at first purchased and assembled together in Area 1 and then ABCD will be transported to Area 2;

ABCD will be assembled with E and F to produce the final product in Area 2.

As shown in Fig. 3.3(b), subassemblies are transported only once between the two areas. If the transportation cost between the two areas is a significant cost element, assembly Plan II has a lower total supply chain cost than assembly Plan I.



(a) Assembly Plan I (b) Assembly Plan II

Figure 2.3. Transportation flows of the two assembly plans.

Under the specified supply chain conditions, Plan II outperforms Plan I in terms of the total supply chain cost. This example demonstrates that supply chain configurations play an important role in optimizing assembly plans for the total cost. However, little research has been conducted on the interactions between assembly plans and supply chain configurations.

2.3 General Problem Formulation and Analysis

The addressed problem will be formulated as a general optimization problem, followed by the identification of the challenges in the problem formulation.

2.3.1 Problem Formulation

Assume we need to make a product, with N components, which can be purchased from suppliers located in different areas. Denote the component supplier set for component i as $\mathbf{S}_i$. To assemble the product out of components, K assembly processes are required, which could be assigned to various manufacturers. The set of suppliers for assembly process j is denoted as $\mathbf{Q}_j$. Since assembly processes have to satisfy some constraints due to manufacturing issues, the sequence of the K assembly processes has to belong to a feasible assembly plan, the set of which is denoted as $\mathbf{AP}$. Each supplier, a component supplier or a manufacturing process supplier, offers a price and a lead time for the service it provides. In this research average time and cost per product or subassembly is used, thus our work could be applied to companies using different strategies, either build-to-order or make-to-stock.

The problem is to decide which assembly plan to use and accordingly how to assign the components and assembly work to suppliers. Depending on the assembly plan and the supplier assignment, components and subassemblies are transported from upstream suppliers to downstream suppliers and/or the assemblers. The transportation cost and time are determined by the locations of the two suppliers related. The objective is to find an assembly plan and supplier assignment so that the product could be assembled at the lowest cost. The cost includes component purchasing cost, assembly cost, inventory cost, and transportation cost. The lead time includes procurement time, assembly time, inventory time, and transportation time.

For a given assembly plan ap belonging to $\mathbf{AP}$, component suppliers vector $\mathbf{s}$, where s_i is the supplier for component i, and assembly suppliers vector $\mathbf{q}$, where q_j is the supplier for assembly process j, define c as the cost function related to assembly process j, pc as the purchasing cost function of component, and d as the delay function related to assembly process j, then the addressed problem could be formulized as:

$$\min\left(\sum_{j=1}^{K} c(j \mid \mathbf{q}, \mathbf{s}, ap) + \sum_{i=1}^{N} pc(i, s_i)\right)$$

Subject to:

$$ap \in \mathbf{AP}, s_i \in \mathbf{S}_i, q_j \in \mathbf{Q}_j$$

Two major challenges exist in this addressed problem. The first one is on the expressions of cost and time using the decision variables, i.e., the functions c and d. The next one is on assembly representation, since different assembly representation may lead to very different search spaces of assembly plans and the cost and time functions could be totally different depending on assembly representations. Fully exploiting the structure of the problem is hence a major strategy to reduce the computational work when approaching the problem using an exact algorithm. Hence various representations will be reviewed first and their complexity and applicability to this problem will be evaluated before a new algorithm is developed for the coordinated optimization problem.

2.3.2 Assembly Representation Analysis

Significant research has been carried out in assembly representation. Most of representations are based on or included in the following three methods: assembly sequences, subassembly state diagram, and AND/OR graph. The focus of this section is placed on the analyses and comparison of these three representations.

Assembly sequence (AS) is an intuitive and a frequently used way to represent the sequences of assembly steps. Typically assembly sequence is represented with a series of indices of liaisons, in which the leftmost liaison has to be accomplished first and the rightmost one will be done finally. The advantage of assembly sequence is its ease for description, which makes the modeling very simple and intuitive. However, assembly sequences cannot reveal the nonlinear relations between assembly processes because a one-dimensional representation cannot represent parallel relations. With a simple enumeration of assembly sequences instead of representing them in a structural way, the number of assembly sequences could be huge.

Assembly State Diagram (ASD) is a directed diagram representing the state of fulfillment of the assembly processes. Every path from the starting node to the ending node is a feasible assembly plan. Then the time and cost required for the whole assembly could be formulated as the accumulated time and cost along the path from the starting node to the ending node. Hence the problem could be modeled as a Shortest Path Problem (SPP).

However, the ASD also has the problem of representing nonlinear relations because an ASD in essence merges assembly sequences in one graph.

The AND/OR graph (AOG) shows the ways to build an assembly step by step from components. It contains two types of logic, i.e., AND logic and OR logic. An AND logic represents what a subassembly is composed of and an OR logic represents different ways of the composition. The AOG possesses the capability to represent nonlinear relations with a tree based structure, i.e., two subassemblies not connected in AOG could be done simultaneously. Since the vertices in an AOG are subassemblies, it is easy to integrate supplier information. The disadvantage of AOG over the other two is its complexity for modeling as it has two types of logic.

The most important feature of a representations method is the size of the models. We follow the definitions of strongly connected assemblies and weakly connected assemblies by Homem De Mello and Sanderson [7] to illustrate the sizes of models of the three representation methods. The sizes of the three representations for strongly connected assemblies are $\left(\frac{n(n-1)}{2}\right)!$, $partions(n)$, and $2^n - 1$. The sizes in the three representations for weakly connected assemblies are $(n-1)!$, 2^{n-1}, and $\frac{n(n+1)}{2}$, where n represents the number of components in the product. Table 2.1 provides a numerical comparison between these three representations.

Table 2.1. Size of models using the three representation methods

	Strongly Connected Assemblies			Weakly Connected Assemblies		
n	AS	ASD	AOG	AS	ASD	AOG
1	1	1	1	1	1	1
2	1	2	3	1	2	3
3	6	5	7	2	4	6
4	720	15	15	6	8	10
5	3628800	52	31	24	16	15
6	1.30e+12	203	63	120	32	21
7	5.10e+19	877	127	720	64	28
8	3.04e+29	4140	255	5040	128	36
9	3.72e+41	21147	511	40320	256	45
10	1.19e+56	115975	1023	362880	512	55
11	1.26e+73	678570	2047	3628800	1024	66
12	5.44e+92	4213587	4095	39916800	2048	78
13	1.13e+115	27644437	8191	479001600	4096	91
14	1.35e+140	190899322	16383	6227020800	8192	105
15	1.08e+168	1382958545	32767	87178291200	16384	120

Four features of the three assembly representation methods in addition to the size of the models are compared as listed below:

1. The complexity to build models;
2. The capability to represent nonlinear relations;
3. The degree of difficulty in supplier information integration;
4. The degree of difficulty in total cost and time reflection.

The comparisons in the five aspects are summarized in the table below.

Table 2.2. Comparison of three assembly representation methods

Representations	Modeling Size	Modeling Complexity	Nonlinear Relation	Supplier Integration	Cost & Time Reflection
AS	Large	Small	No	High	High
ASD	Medium	Medium	No	High	Low
AOG	Small	Large	Yes	Low	Low

Although AOG has a relatively high complexity for modeling, its capability to represent nonlinear relations, potential to integrate supply chain, and small modeling size make it a preferred assembly modeling option. Hence AOG is chosen for assembly representation in this thesis. However ASD might also find its use in assembly modeling for the addressed problem due to its merit in modeling the optimization problem as a SPP. Other researchers may find it promising to develop an algorithm based on ASD.

2.4 Mathematical Modeling

This section provides answers to the questions identified in the problem analysis section, i.e., how to represent assembly plan and supply chain cohesively as well as how to derive the cost and lead time functions.

2.4.1 Hyper AND/OR Graph

Here we proposed a new concept, the Hyper AND/OR graph, or HAG. In general, AHAG is defined as:

$\mathbf{H} = (\mathbf{V}, \mathbf{A}, \mathbf{E})$, where $\mathbf{V}$ is the set of vertices, $\mathbf{A}$ is a set of triples of vertices, and $\mathbf{E}$ is a set of non-empty subsets of $\mathbf{V}$ called hyper edges.

Define $\mathbf{C}$ as the set of components of product, i.e. $\mathbf{C} = \{c_1, c_2, \ldots c_N\}$. Then $\mathbf{V}$ is a subset of $\mathrm{P}(\mathbf{C}) \setminus \{\phi\}$, i.e. $\mathbf{V} \subseteq \mathrm{P}(\mathbf{C}) \setminus \{\phi\}$, where $\mathrm{P}(\mathbf{C})$ is the power set of $\mathbf{C}$. To make each element of $\mathbf{V}$ a feasible subassembly, it has to satisfy the subassembly criterion, i.e. precedence and stability [7].

$\mathbf{V} = \{v_i \mid i \in \mathbf{I_v}\}$, where $\mathbf{I}_v$ is the index set of the vertices. Each v_i represents a feasible subassembly. Vertices with one component are called simple vertices (*SV*) and those with multiple components are called compound vertices (*CV*).

$\mathbf{A} = \{\mathbf{a}_i \mid \mathrm{i} \in \mathbf{I}_a\}$, where $\mathbf{I}_a$ is the index set of the AND relations. And $\mathbf{a}_i = (v_0^i, v_1^i, v_2^i)$, $v_0^i = v_1^i \cup v_2^i$. $\mathbf{a}_i$ represents an AND relation between v_0^i, v_1^i, and v_2^i, meaning subassembly v_0^i is made up with subassemblies v_1^i and v_2^i. v_0^i is called parent vertex, v_1^i called left child vertex, and v_2^i called right child vertex for ease of description although it is unnecessary to differentiate between two child vertices. Define level of vertex as its hierarchy level and that of *SV*'s level 0.

$\mathbf{E} = \{\mathbf{e}_i \mid \mathrm{i} \in \mathbf{I}_e, \mathbf{e}_i \subseteq \mathbf{V}\}$, where $\mathbf{I}_e$ is the index set of the edges. $\mathbf{e}_i$ represents the subassemblies that can be assigned to one supplier. And $|\mathbf{E}|$ is the total number of suppliers.

2.4.2 Supply Chain Modeling

HAG can reflect material and transportation flows through the extra layer of supply chain information incorporated. Given an AND relation $\mathbf{a} = (v_0, v_1, v_2)$, the material and transportation flow could be seen in Figure 2.4 below:

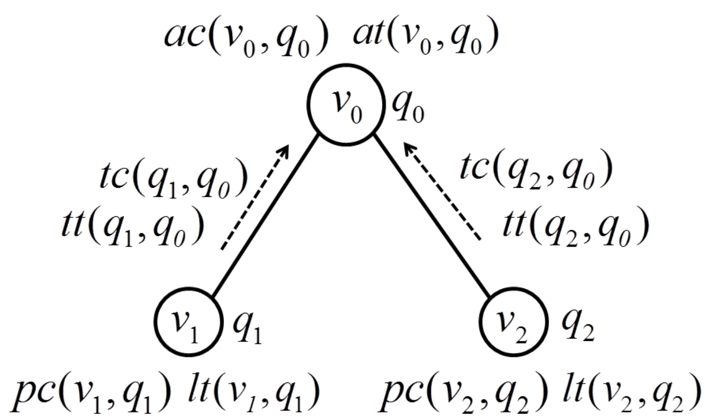


Figure 2.4. Material and transportation flow.

To procure the subassembly through this AND relationship, supplier q_0 needs to purchase subassembly v_1 from supplier q_1 and subassembly v_2 from supplier q_2.

Denote $pc(v,q)$ as the procurement cost of subassembly v from supplier q, $ac(v,q)$ as the assembly cost pertaining to subassembly v from assembly process provider q, and $tc(q_1, q_2)$, as transportation cost between suppliers q_1 and q_2.

Then the purchasing cost of subassembly v_0 is:

$$pc(v_0, q_0) = pc(v_1, q_1) + tc(q_1, q_0) + ac(v_2, q_2) + tc(q_2, q_0) + ac(v_0, q_0).$$

Similar to the above cost model, lead times include two parts, the assembly lead time as well as transportation lead time. Denote the subassembly procurement lead time $lt(v,q)$, assembly lead time $at(v,q)$, and transportation lead time $tt(s_1,s_2)$. The lead time for a subassembly is determined by the one of its two subassemblies with a longer procurement lead time, i.e.

$$lt(v_0,q_0) = \max(lt(v_1,q_1) + tt(q_1,q_0), lt(v_2,q_2) + tt(q_2,q_0)) + at(v_0,q_0)$$

HAG incorporates assembly information as well as supply chain information in one graph and conveys material and transportation flows between suppliers in a hierarchical way. The remaining challenge is how to calculate the total cost and lead time, which will be tackled in the following chapter.

2.5 Summary

This chapter begins with a simple pen example to illustrate the significance of the integrated designs of assembly plans and supply chains. A formulation of the integrated optimization problem is provided using the most general optimization modeling method. Then two major challenges are identified in the formulation, i.e., assembly representations as well as cost and time calculation. After a review of the three most widely used assembly representation methods, i.e., assembly sequence, assembly state diagram, and AND/OR graph, the possibilities to utilize these three representations in the addressed problem are discussed. AND/OR graph is chosen due to its advantages in model size, supply chain integration, as well as cost and time calculation. Through sharing the commonality among the assembly plans to the most extent, AND/OR graph has a polynomial complexity for weakly connected assembly products. This systematic comparison provides a fundamental understanding of the problem complexity and potential directions to model the problem.

Hyper AND/OR Graph is developed to incorporate supply chain decisions in this dissertation and the mathematical model is also provided. HAG serves as a good platform for research on the integrated decisions on assembly planning and supply chain configurations.

Moreover, HAG may find various applications in a series of Resource Constrained Shortest Path Problems (RCSP) due to its capabilities to incorporate nonlinear constraints and network information. A good example is Vehicle Routing Problem (VRP), which is a combinatorial optimization and integer programming problem that asks "What is the optimal set of routes for a fleet of vehicles to traverse in order to deliver to a given set of customers?" In this case, the sequences between the customers could be represented in the AND/OR graph while the holding capacity could be represented as the hyper nodes. Powered by the efficient representation of HAG, we can now move on to algorithm development.

CHAPTER 3

Algorithm for Integrated Design of Assembly Plan and Supply Chain

Nomenclature

$\Pi(\mathbf{X})$	The set of all subsets of set $\mathbf{X}$
$\lvert \mathbf{X} \rvert$	The cardinality of set $\mathbf{X}$
N	Number of nodes in a liaison graph
K	Number of liaisons in a liaison graph
l	Liaison index, $1 \le l \le K$
i, j	Component index or node index in the hyper AND/OR graph
k	AND relation index
p_i	Node indexed by i in a liaison graph, $1 \le i \le N$
$\mathbf{P}$	Set of nodes in a liaison graph
$\Pi(\mathbf{P})$	Set of all groups of components
M	Number of compound nodes in an AND/OR graph
c_q	Compound nodes indexed by q in an AND/OR graph, $1 \le q \le M$
$\mathbf{C}$	Set of compound nodes, representing subassemblies, in an AND/OR graph
$\mathbf{A}$	Set of nodes in an AND/OR graph $\mathbf{A} = \mathbf{C} \cup \mathbf{P}$
a_i	Node indexed by i in the hyper AND/OR graph, $1 \le i \le M + N$
d_i	The number of AND relations of compound node a_i

	$(N+1 \le i \le N+M)$
H	Composition relationship matrix in a hyper AND/OR graph
s	Supplier of a component or a manufacturing process
$\mathbf{S}_i$	The supplier set of component i, $1 \le i \le N$
$\mathbf{T}_l$	The supplier set of manufacturing process l, $1 \le l \le K$
$\mathbf{Sp}$	The set of all suppliers, $\mathbf{Sp} := \{Sp_j \mid 1 \le j \le S\}$
S	The total number of suppliers
$\mathbf{Hn}$	The set of hyper nodes in the hyper AND/OR graph, $\mathbf{Hn} := \{Hn_j \mid 1 \le j \le S\}$
$pc(i,s)$	The procurement cost of component i from supplier s, $s \in S_i$
$mc(i,s)$	The manufacturing cost of liaison l from candidate processor s, $s \in T_l$
$\mathrm{tic}(s_1, s_2)$	The sum of transportation and inventory costs between suppliers s_1 and s_2, $s_1, s_2 \in Sp$
$\mathrm{lisn}(i,k)$	The liaison index of the k^{th} AND relation of node i, $1 \le i \le M$, $1 \le k \le d_i$
sub	Subassemblies of the product
$fsub$	Final product
$\mathrm{cost}(sub, s)$	Cost of subassembly sub from supplier s
(i,k,s)	State of the DP procedure
$V(i,k,s)$	Value function of the DP procedure
V	The optimal cost of the final product
$\mathrm{child}_1(i,k)$	The left child node index of node i under AND relation k

$\text{child}_2(i,k)$	The right child node index of node *i* under AND relation *k*
$\text{opt}_1(i,k,s)$	The optimal solution of the left child node of node *i* under AND relation *k* with supplier choice *s*
$\text{opt}_2(i,k,s)$	The optimal solution of the right child node of node *i* under AND relation *k* with supplier choice *s*

3.1 Introduction

This chapter focuses on the algorithm development for the integrated optimization without time constraints based on the models introduced in the previous chapter. First we will review the related literature on concurrent designs. After the identification of research gaps and barriers, we will introduce the solution algorithms and provide two case studies, one on Boujault's pen and the other on a laptop computer.

In recent years, researchers began to consider manufacturing factors in supply chain designs. Rungtusanatham and Forza [29, 30] summarized such efforts in integrating decision makings in design, manufacturing, and supply chains. Huang, Zhang, and Liang [31] considered the manufacturing processes in the supply chain configuration for a product family by applying a genetic algorithm. Fine, Golany, and Naseraldin [32] evaluated a tradeoff between the design of the product, manufacturing process, and supply chain through a goal programming approach. Fixson [33] assessed product architecture by building a tool to link product, process, and supply chain decisions.

In the aforementioned work, manufacturing processes were only considered as an option choice, while how assembly sequences and/or subassembly definition could affect product qualities or costs were not studied. Shao et al. [34] developed a new integration model with a modified GA-based approach to facilitate the integration and optimization of process planning and scheduling systems. Che [35] presented a mathematical model to deal with production planning problem of selecting assembly sequences and suppliers. A hybrid heuristic algorithm, Guided-Pareto genetic algorithm (*Gu-PGA*), was developed to minimize the integrated criteria. Che and Chiang [36] integrated supplier selection, product assembly, as well as the logistic distribution system of the supply chain for the build-to-order supply chain planning. A Pareto genetic algorithm (*PaGA*) was developed to find good tradeoffs among three evaluation criteria, namely costs, delivery time, and quality. These work provided methods to concurrently select assembly sequences and supply chain configurations by using heuristic algorithms. However, due to simply enumerating the assembly sequences, the optimization spaces were immensely large, even for problems with moderate sizes. Moreover, the relations between assembly and supply chain have not yet been revealed. This chapter aims at developing a complete and efficient algorithm to solve the integrated design problem.

3.2 Problem Formulation

This section provides a formal problem formulation based on HAG and the supply chain modeling introduced in Chapter 2.

Assume we are producing a product $\mathbf{P} = (\mathbf{C}, \mathbf{L})$, with N components and K assembly processes, where $\mathbf{C}$ is the set of components of the product, i.e., $\mathbf{C} = \{c_1, c_2, \ldots c_N\}$ and $\mathbf{L}$ is the set of assembly processes, i.e. $\mathbf{L} = \{l_1, l_2, \ldots l_K\}$. The set of suppliers for component c_i is denoted as $\mathbf{S}_i, i \le N$ and the set of providers of each assembly process is denoted as $\mathbf{Q}_j, j \le \mathrm{K}$. Please note that some suppliers could provide components as well as assembly processes, i.e. $\mathbf{S}$ and $\mathbf{Q}$ are not mutually exclusive. The purchasing cost for component i from supplier $s \in \mathbf{S}_i$ is $pc(i,s)$. The assembly cost for assembly process j from supplier $q \in \mathbf{Q}_j$ is $ac(j,q)$, and the corresponding assembly lead time is $at(j,q)$. While purchasing of components is the only action needed for simple vertices, the decisions of where to assign the assembly processes and subassemblies need to be made for compound vertices.

Then the HAG for this product $\mathbf{H} = (\mathbf{V}, \mathbf{A}, \mathbf{E})$ is as follows:

$\mathbf{V} = \{v_i \mid i \in \mathbf{I_v}\}$, is the set of feasible subassemblies, where each v_i represents a subassembly. The number of vertices in $\mathbf{H}$ is $|\mathbf{I_v}|$. Define $M = |\mathbf{I_v}| - N$, i.e. the number of CV's in $\mathbf{H}$ is M , then the total number of vertices in $\mathbf{H}$ is $M + N$. For ease of description, we re-label vertices from low levels to high levels in an ascending order using function $leb(v)$ so that $i > j$ if the level of v_i is higher than that of v_j . Hence the SV's are labeled from 1 to N , while CV's are labeled from $N+1$ to $M + N$.

$\mathbf{A} = \{\mathbf{a}_i \mid i \in \mathbf{I_a}\}$, where $\mathbf{a}_i = (v_0^i, v_1^i, v_2^i)$. Dependent on the parent vertex of each AND relation, $\mathbf{A}$ is partitioned into M subsets, i.e. $\mathbf{A} = \bigcup_{i=1}^{M} \mathbf{A}_i$, where $\mathbf{A}_i = \{\mathbf{a}_j \mid leb(v_0^j)$

$= i + N, j \in \mathbf{I_a}\}$. The assembly process required to assemble v_1^i and v_2^i into v_0^i could be indexed by function $fapi(i)$. Then CV v_{N+i} has $|\mathbf{A}_i|$ AND relations with the set $\mathbf{A}_i$. For each of its AND relationship $\mathbf{a}_j = (v_0^j, v_1^j, v_2^j)$, the two subassemblies of this CV are v_1^j and v_2^j, where $j \in \mathbf{A}_i$.

$\mathbf{E} = \{\mathbf{e}_i \mid i \in \mathbf{I_e}, \mathbf{e}_i \subseteq \mathbf{V}\}$. $|\mathbf{E}|$ is the total number of suppliers, including both component suppliers and assembly suppliers, which is also denoted as S. The supplier set for SV $v_i, i \in [1, N]$ is $\mathbf{S}_i$ .The corresponding assembly process to AND relationship $\mathbf{a}_j = (v_0^j, v_1^j, v_2^j)$ of CV v_{N+i} is $fapi(j)$ with an assembly supplier set $\mathbf{Q}_l$, where $l = fapi(j)$. Then the hyper edge pertaining to i^{th} supplier s is $\mathbf{e}_i = \{\mathrm{v}_j \mid \mathrm{s} \in \mathbf{S}_j \text{ or } \mathrm{s} \in \mathbf{Q}_l\}$, where $l = fapi(j)$.

Hence the objective is to minimize the cost of the final vertex given the assembly and supply chain resources, i.e.

$$\min_{\mathbf{a},q}(pc(v_{M+N}, q \mid \mathbf{a})), \text{ where } \mathbf{a} \in \mathbf{A}_{M+N}, \text{ and } \mathrm{q} \in \mathbf{Q}_l \text{ in which } l = fapi(\mathbf{a}).$$

Three types of decisions have to be made recursively for each compound vertex,

1. Choose an AND relation, i.e. j, among $|\mathrm{A}_i|$ AND relations,
2. Select a supplier for the assembly process, i.e., q, among $|\mathbf{Q}_l|$, where $l = fapi(j)$,
3. Decide where to obtain its two subassemblies v_j^1, v_j^2.

Formulated in this way, this problem is decomposed into two sub-problems, i.e., how to obtain each of its two subassemblies at low costs. Dynamic Programming (DP) is selected for the modeling of the optimal configuration because of its merits in structured search. A bottom-up DP algorithm is designed to make the assembly plan and supply chain decisions considering the fact that the supply chain cost is cumulative.

3.3 Algorithm Development and Analysis

Based on the mathematical model in the previous section, the DP algorithm to the coordinated optimization problem without time constraints is firstly introduced, and then its computational complexity is analyzed.

3.3.1 DP Algorithm Description

For ease of the algorithm description, some definitions are given as follows. Define $d_i = |\mathbf{A}_i|$, i.e. d_i is the number of possible ways to assemble subassembly v_i. Define $\mathbf{b}_i$ as the array of indices of AND relations belonging to subassembly v_i, i.e. $\mathbf{b}_i(\mathrm{k}) = j$ so that $leb(v_0^j) = i + N$, $j \in \mathbf{I_a}$.

State: (i,k,s), where i is the index of a compound vertex in the hyper AND/OR graph ($N+1 \le i \le N+M$), k is the AND relation index of vertex i ($k \le d_i$), and s is the supplier of subassembly i.

Actions: For each AND relation, i.e., for each way to build a compound vertex and the corresponding assembly process supplier, decide where to acquire its two subassemblies.

Value function: The total supply chain cost to produce compound vertex i is denoted as $V(i,k,s)$, given the assembly type, i.e., the AND relation index $k \le d_i$, as well as the corresponding manufacturing process provider $s \in S_l$ for process l, where $l = \text{lisn}(i,k)$.

Functional equation: Compound vertex i makes decisions on where to acquire its two child vertices to minimize its value function at the state (i,k,s). The optimization process is described below.

The two subassemblies (vertices) of vertex i at state (i,k,s) are v_1^j and v_2^j, where $\mathbf{b}_i(\text{k}) = j$. We denote j_1 as the left child vertex and j_2 as the right child vertex, i.e. $j_1 = \text{child}_1(i,k)$ and $j_2 = \text{child}_2(i,k)$.

Subassembly (vertex) j_1 has d_{j_1} AND relations. The liaison to accomplish and for each assembly method $k_1 \le d_{j_1}$ is $l_1 = \text{lisn}(j_1,k_1)$. The supplier for subassembly j_1 is denoted as s_1 and $s_1 \in S_{l_1}$. The costs related to vertex j_1 include procurement of j_1 from supplier s_1 with the cost of $pc(s_1)$ as well as the sum of the cost of transportation from supplier s_1 to supplier s, $\text{tc}(s,s_1)$.

For subassembly (vertex) j_2, similarly, the procurement cost is $pc(s_2)$, $s_2 \in S_{l_2}$, $l_2 = \text{lisn}(j_2,k_2)$, $k_2 \le d_{j_2}$ and the transportation cost $\text{tc}(s,s_2)$.

Hence, the value function of vertex i at state (i,k,s) can be calculated as below:

$$V(i,k,s) = \min_{k_1} \min_{s_1}(V(j_1,k_1,s_1) + \text{tc}(s,s_1)) + \min_{k_2} \min_{s_2}(V(j_2,k_2,s_2) + \text{tc}(s,s_2)) + mc(s)$$

where $k_1 \le d_{j_1}$, $l_1 = \text{lisn}(j_1, k_1)$, $s_1 \in S_{l_1}$; $k_2 \le d_{j_2}$, $l_2 = \text{lisn}(j_2, k_2)$, $s_2 \in S_{l_2}$; $k \le d_i$, $l = \text{lisn}(i,k)$, $s \in S_l$.

The minima is recorded to trace back the assembly plan, i.e.,

$$\text{opt}_1(i,k,s) = (k_1^*, s_1^*) \text{ and } \text{opt}_2(i,k,s) = (k_2^*, s_2^*).$$

Boundary condition: while compound vertices in the HAG are associated with decisions of assembly plans and subassembly suppliers, the only decision associated with a simple vertex is where to purchase the component. The value function of a simple vertex is the purchasing cost of the component.

$V(i,k,s) = pc(s), k = 1, s \in S_i$, if $i \le N$.

Optimal Result: when $i = M + N$, the minimal total supply chain cost V can be found by comparing the d_i costs provided by d_i optimized assembly plans with the corresponding supplier chain configurations, i.e. $V = \min_{k,s} V(i,k,s)$. The optimal assembly plan and supply chain configurations can be obtained via a top-down tracking of $\text{opt}_1(i,k,s)$ and $\text{opt}_2(i,k,s)$.

3.3.2 Computational Complexity Analysis

As shown in the functional equation, the number of computations for the optimization at state (i,k,s) is $\sum_{k_1=1}^{d_{j_1}} |S_{l_1}| + \sum_{k_2=1}^{d_{j_2}} |S_{l_2}|$, where $j_1 = \text{child}_1(i,k)$, $j_2 = \text{child}_2(i,k)$, $l_1 = \text{lisn}(j_1, k_1)$ and $l_2 = \text{lisn}(j_2, k_2)$. Then the total number of computations for the complete optimization problem is $\sum_{i=N+1}^{N+M}\sum_{k=1}^{d_i}\sum_{s=1}^{|S_l|}(\sum_{k_1=1}^{d_{j_1}}|S_{l_1}|+\sum_{k_2=1}^{d_{j_2}}|S_{l_2}|)$, where $l=\text{lisn}(i,k)$. Theoretically, d_i, d_{j_1}, and d_{j_2} have positive correlation to the vertex index i, as complicated subassemblies usually can be assembled in more ways. We can calculate the theoretical computation complexity of this DP algorithm by studying the relationships between vertex index i and its number of AND relations (i.e. d_i, d_{j_1}, and d_{j_2}). However the exact values of these numbers depend on assembly structures and it is not the focus of this research. A practical upper bound is provided instead for the computational complexity of the algorithm which can be utilized to estimate the performance of the algorithm running for a practical product. Define D as the maximum number of possible ways to assemble a subassembly, and S as the maximum number of suppliers for a component or a manufacturing process. Then one upper bound of the computational complexity of the algorithm is $O(MD^2S^2)$, where M is the number of compound vertices in the hyper AND/OR graph. In typical products, $M=O(N^2)$ and $D=O(N)$, and then the upper bound of the computational complexity is $O(N^4S^2)$. In contrast, the enumeration has a computational complexity $O(KNS^{L+N})$, where K is the number of assembly plans, L is the number of processes required, N is the number of components in the product, and S is the maximum number of suppliers for a component or a process. In typical products, $K=O(N^2)$ and $L=O(N)$, and the computational complexity is $O(N^2S^{2N})$. By taking advantage of the compactness of the proposed hyper AND/OR graph and searching in a structured way, the DP algorithm solves the optimization problem with an exponential time complexity in polynomial computational time.

3.4 Case Studies

Two case studies are provided in this section to illustrate the efficacy and efficiency of the developed algorithm.

3.4.1 Pen Example Case Study

The discussion on the pen example will be completed in this section. There are six assembly plans in total to produce the pen discussed in Chapter 2. The assembly plans with the optimized supply chain configurations and the corresponding optimal costs are presented in Figure 3.1.

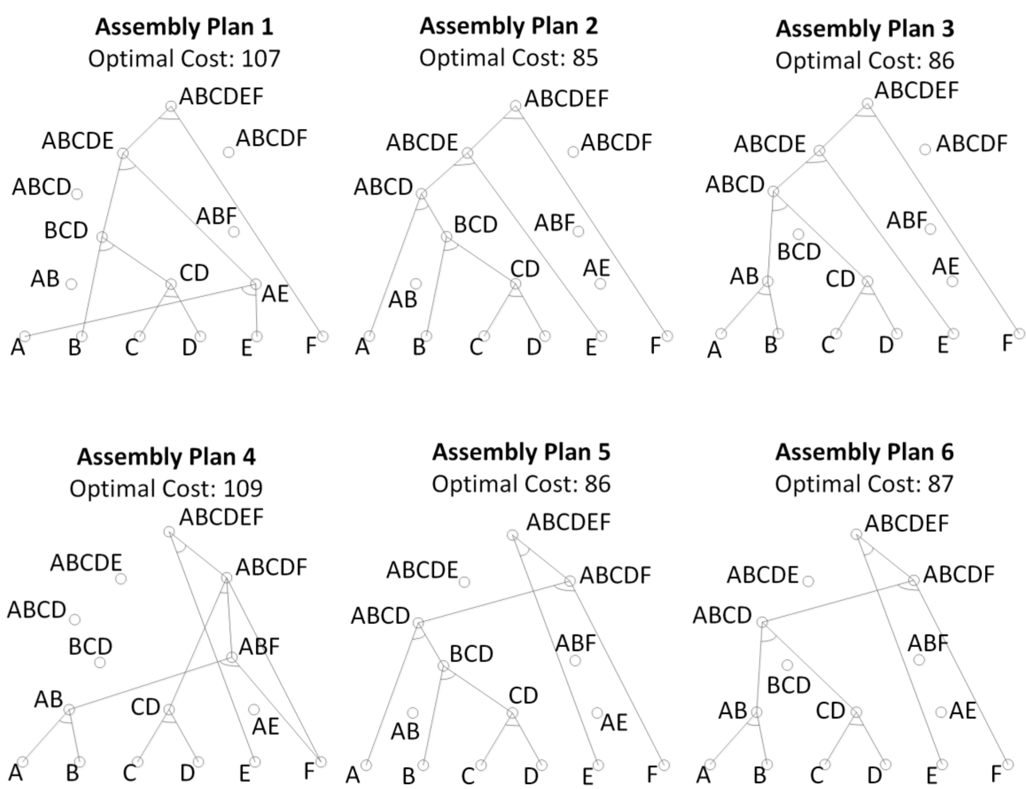


Figure 3.1. Assembly plans of the pen

The six assembly plans can be categorized into two groups based on the total supply chain cost. Assembly plans 2, 3, 5, and 6, have optimal cost around 86, while assembly plans 1 and 4 have costs around 108, around 20% higher than the other four plans. All the "good" assembly plans include the "ABCD" subassembly while the "bad" assembly plans do not. The result shows it is cost-effective to produce "ABCD" before transporting components or subassemblies to Area 2. As illustrated in Chapter 1, the supply chain configuration characteristic, i.e., the long distance between Areas 1 and 2, leads to the result.

Through a learning tool such as a decision tree or boosting learning algorithms, the proposed method can provide guidelines on which intermediate products should be selected as subassemblies in order to keep the supply chain cost-effective. These guidelines can guide assembly designers in their designs of assembly plans as well as supply chain configurations.

3.4.2 Laptop Computer Case Study

The developed method is applied to a laptop assembly to evaluate how the method works for a practical product. The laptop case is adapted from a study [37]. Since we are only concerned with liaisons that require assembly processes, we do not consider the contact liaisons [4] denoted in dashed lines in Figure 3.2.

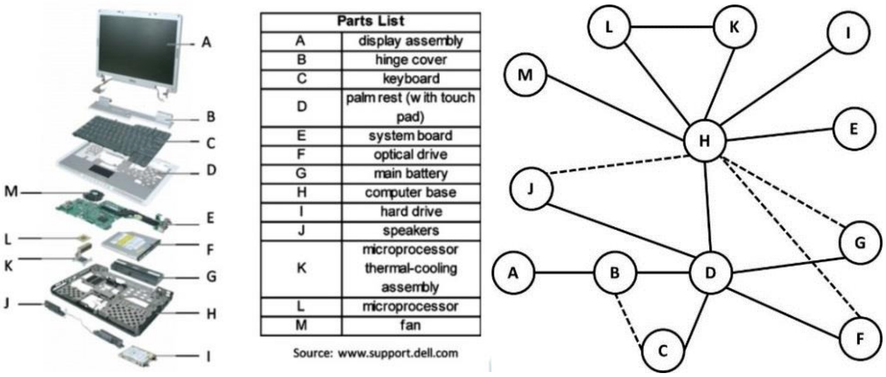


Figure 3.2. Laptop and the liaison graph, adapted from [37]

The precedence graph in Figure 3.3 remains the same as that in that paper. The AND/OR graph of the laptop can be generated based on algorithms in [9]. There are a total of 5,280 assembly plans in the hyper AND/OR graph of the laptop. Due to such a huge number and complicated graph, we include only a part of the AND/OR graph in Figure 3.4.

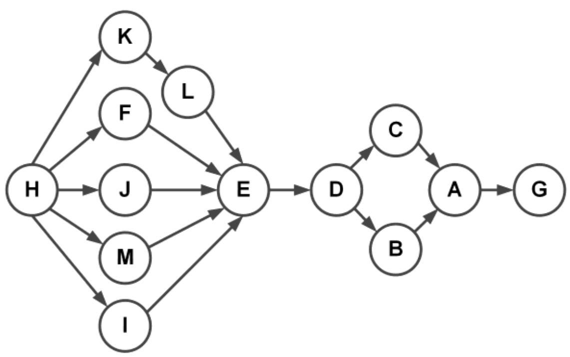
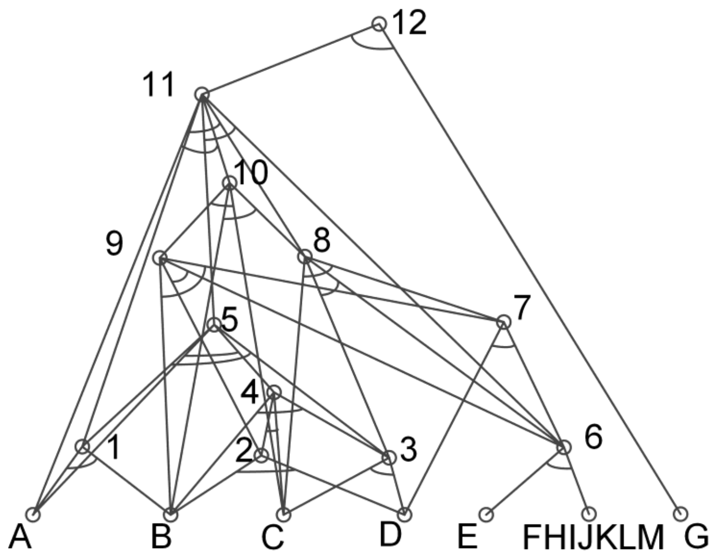


Figure 3.3. Precedence graph of the laptop, adapted from [37]

Assume a simplified but realistic supply chain environment, in which suppliers are located in four districts. Most components and assembly processes could be obtained in District 1, while specific components or processes could be acquired in the other districts. The details could be found in. The numbers in the table are generated by modifying publically available estimated data.



Name List	
1	AB
2	BD
3	CD
4	BCD
5	ABCD
6	EFHIKLM
7	DEFHIKLM
8	CDEFHIKLM
9	BDEFHIKLM
10	BCDEFHIKLM
11	ABCDEFHIKLM
12	ABCDEFGHIKLM

Figure 3.4. A part of the AND/OR graph of the laptop

District 1 and 3 can provide all the assembly processes, while District 2 has specialty in assembly work related to display. There are 52 suppliers, providers of components or assembly processes, in the supply chain. If the two suppliers are in the same district, the transportation cost is fairly low, roughly \$1. Otherwise if they are companies located in different districts of the first three districts, the transportation cost is high, around \$6. If the transportation is between District 4 and the other districts, the cost is even higher, around \$10.

Table 3.1. Component availability and prices in four districts

Region / Component	District 1	District 2	District 3	District 4
A	80	70	×	×
B	12	8	×	×
C	30	35	×	×
D	10	×	×	×
E	85	×	85	85
F	35	×	40	×
G	×	×	×	40
H	10	×	×	×
I	55	×	54	×
J	8	×	×	×
K	5	×	4	×
L	160	×	150	155
M	3	×	×	×

The optimal assembly and supply chain plan generated by the proposed method is shown in Figure 3.5 and the corresponding material flow in Figure 3.6.

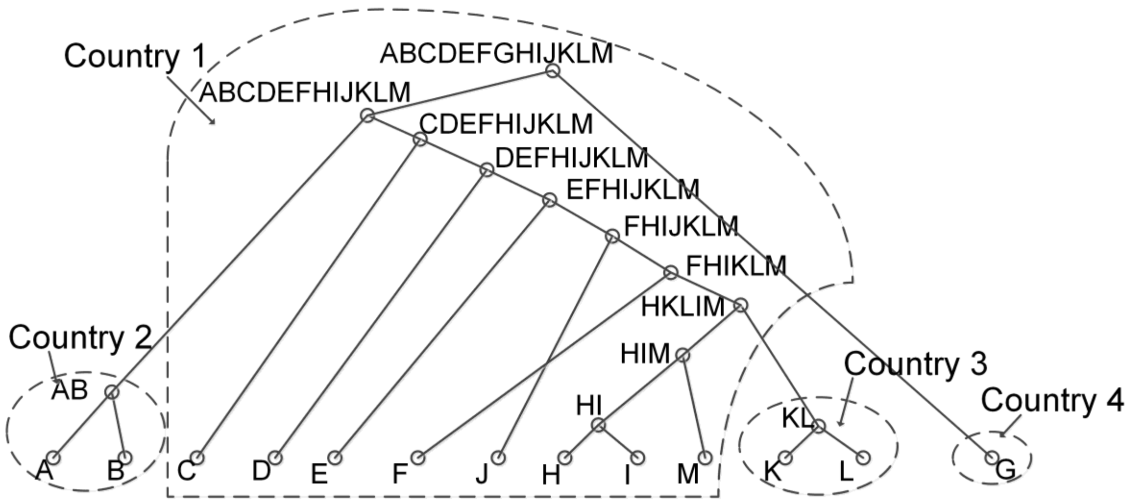


Figure 3.5. The optimal assembly plan with supplier assignment

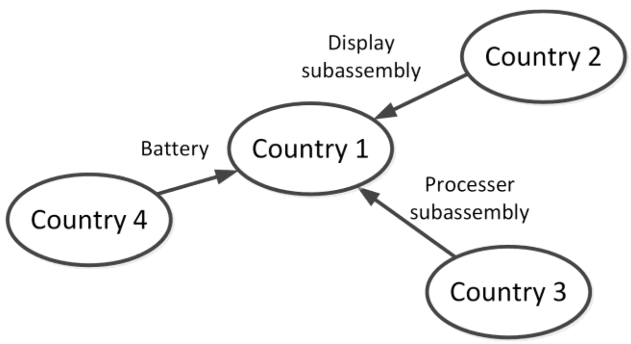


Figure 3.6. The material flow of the optimal assembly plan and supply chain

Companies in District 2 are assigned the subassembly of display due to its expertise in display. Companies in District 3 are assigned the processor subassembly because of their advantages in processor production. The battery is provided by District 4. All the other components are purchased and then assembled in District 1. Companies in District 1 are chosen for the most assembly work, although they offer the assembly processes at a higher price than those in District 3. That is because they could provide the parts such as palm rest and computer base so that there will be much less transportation of subassemblies between districts. This result demonstrates why industrial park is a competitive strategy.

Assume that District 2 can no longer provide a competitive price for display, which rises from \$70 to \$80. The optimal assembly plan with supplier assignment is shown in Figure 3.7. Losing the advantage of the lower price of the display, AB is not chosen as a subassembly, and instead ABCD is selected as a subassembly manufactured in District 1. The subassembly EFHIJKLM, as we could notice, does not change its supplier assignment.

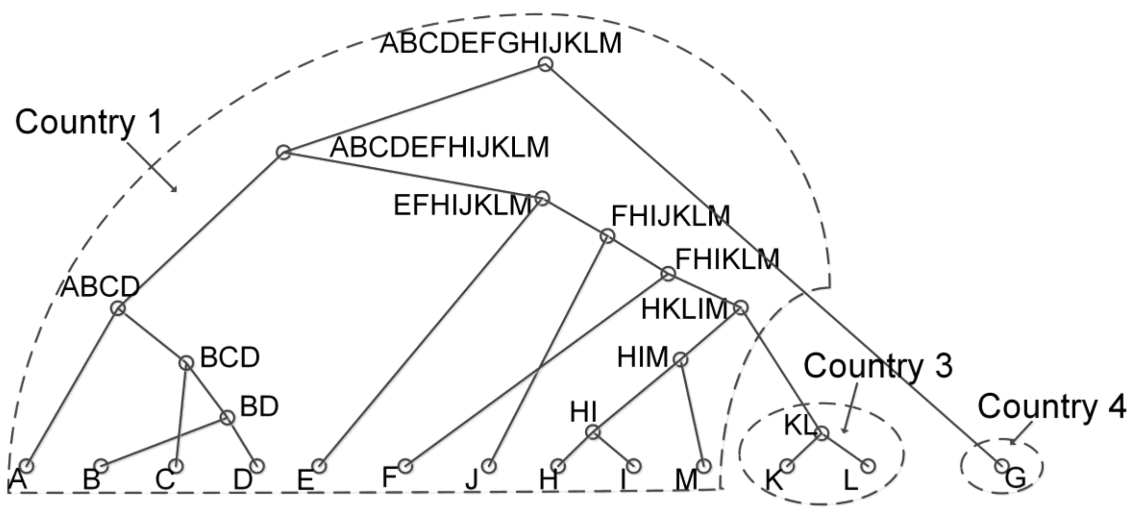


Figure 3.7. The optimal configuration: with increased display price in District 2

In summary, the model can successfully and efficiently make integrated decisions in assembly planning and supplier assignment for the case study product under different scenarios. And the results can give designers suggestions on how to integrate design of assembly plans and supply chain configurations.

3.5 Summary

A bottom-up DP algorithm with polynomial time complexity has been developed to search for the global optimal assembly plans and supply chain configurations with the lowest total cost. The effectiveness and efficiency of the method has been demonstrated through a case study of a laptop computer assembly. Industry cluster strategy was verified to be a successful one through the laptop case study. The DP algorithm solves the integrated decision problem with polynomial time complexity in terms of the model parameters. Through better understanding of the interdependence between suppliers and assembly plans, concurrent decisions can be made to lower the total supply chain cost and shorten the product development lead time.

The developed model can be extended to address other problems such as the effects of the integrated decisions on product lead time, manufacturing reliability, and product reconfiguration. An example as we discussed in the modeling chapter is VRP, which is a NP-hard problem. The commercial solvers therefore tend to use heuristic due to the size of real world VRPs and the frequency that they may be used. The DP algorithm proposed in the dissertation provides a good platform to exercise the heuristics due to its advantages to decompose a problem into smaller problems.

Utilized in combination with Enterprise Resource Planning (ERP), the method developed in this dissertation can see various applications in manufacturing industries. The companies can use the method to generate preliminary manufacturing and supply chain designs based on history data and initial product design before launching a new product. This preliminary design serves as a baseline for further improvement or revision from experts. When a new generation of products is introduced, the method can be run for a part of the product to generate the optimal configurations. The partial optimization allows for quick iterations and evolvement between generations of products. The same technique of partial optimization also yields huge benefits when supply chain fluctuation happens. Under that situation, quick adjustment can be made based on partial optimization results without changing the whole configurations.

CHAPTER 4

Algorithm for Integrated Design Considering Lead Time Constraint

Nomenclature

N	The number of components
K	The number of assembly processes
T	The lead time bound
$\mathbf{C}$	The set of components
c_i	Component indexed by i
$\mathbf{L}$	The set of assembly processes
l_k	Assembly process indexed by k
$\mathbf{S}_i$	The set of suppliers for component i
$\mathbf{Q}_k$	The set of suppliers for assembly process k
s	Supplier for component
$\mathbf{s}$	Supplier array for all the components, where s_i is the supplier index for component i
q	Supplier for assembly processes
$\mathbf{q}$	Supplier array for all the components, where q_k is the supplier index for assembly process k
ap	A feasible assembly plan
$\mathbf{AP}$	The set of feasible assembly plans

$pc(i,s)$	The purchasing cost of component i from supplier s
$c(j\,\|\,\mathbf{q},\mathbf{s},ap)$	The cost related to assembly process j given the supplier selection $\mathbf{s}$ and $\mathbf{q}$, as well as the assembly plan ap.
$d(j\,\|\,\mathbf{q},\mathbf{s},ap)$	The lead time related to assembly process j given the supplier selection $\mathbf{s}$ and $\mathbf{q}$, as well as the assembly plan ap.
$ac(j,q)$	The assembly cost of assembly process j from supplier q
$at(j,q)$	The assembly time of assembly process j at supplier q
v_i	Vertex indexed by i
$\mathbf{V}$	The set of vertices in a hyper AND/OR graph
$\mathbf{a}_i$	AND relation indexed by i
v_p^i, v_l^i, v_r^i	The parent vertex, left child, and right child pertaining to AND relation i
$\mathbf{A}$	The set of all AND relations
$\mathbf{A}_i$	The set of AND relations pertaining to vertex i
$leb(v)$	The label of vertex v
$\mathbf{E}$	The set of hyper edges
$\mathbf{e}_i$	Hyper edge indexed by i
$fapi(j)$	The function for calculating the index of an assembly process for AND relation j
$\mathbf{H}=(\mathbf{V},\mathbf{A},\mathbf{E})$	A hyper AND/OR graph
$d_i = \|\mathbf{A}_i\|$	The number of AND relations pertaining to vertex i
$\mathbf{b}_i(\mathrm{k}) = j$	The array of AND relation indices pertaining to vertex i
i_1, i_2	Indices for vertices
k_1, k_2	Indices for AND relations

l_1, l_2	Indices for assembly processes
s_1, s_2	Indices for suppliers
$tc(v, s_1, s_2)$	The average transportation cost of assembly v between two suppliers
$tt(v, s_1, s_2)$	The average transportation time of assembly v between two suppliers

4.1 Introduction

This chapter focuses on the algorithm development of the integrated optimization problem considering the total lead time. This problem belongs to the category of Resource Constrained Optimization Problem (RCOP). Hence this chapter starts a literature review on RCOP. Then the formal algorithm is introduced and discussed in terms of computational performance. Numerical studies are also presented to demonstrate the efficacy and computational efficiency of the algorithm.

Resource Constrained Scheduling Problem (RCSP) is a general scheduling problem on optimizing the objective function given limited total resources. RCSP contains many variations, the most important of which are Resource Constrained Project Scheduling Problem (RCPSP) and Resource Constrained Shortest Path Problem (RCSPP). Since RCSP are generally NP-hard, as proven in [38, 39], the algorithms for RCSP are usually categorized into two groups, i.e., exact algorithms for small scale problem, and approximation algorithms for medium or large scale problem, as summarized in [40]. The early attempts could date back to 1960s, when Joksch [41] approached RCSPP through linear programming and dynamic programming with a conclusion that exact algorithm didn't have much advantage over a naïve exhaustive search. Due to the practical importance of RCPSP, many exact algorithms with exponential time complexity have been proposed. Early examples are dynamic programming approaches by Hindelang and Muth [42] and Robinson [43]. The currently best known algorithms still rely on dynamic programming, but exploit in addition the decomposition structure of the underlying network. The decomposition that facilitates the computation is known as modular decomposition or substitution decomposition and has many applications in network and other combinatorial optimization problems see the comprehensive article by Mohring and Radermacher [44]. Feillet et al. developed an exact algorithm based on Desrochers' label correcting algorithm for some Vehicle Routing Problems (VRP) and the algorithm was shown an experimentally efficient feature [45].

Various approximation algorithms have been developed through different perspectives. The early attempts were mainly on heuristic rules. Davis and Patterson [46] ran 8 different heuristic rules on 83 problems and found out that none of the heuristic rules tested performed consistently best on all RCPSP problems. Handler and Zang [47] utilized a kth shortest path algorithm and a Lagrangian relaxation to reduce the value of k in order to save computational works. Hartman and Kolisch underwent an experimental evaluation of heuristics for RCPSP in two directions, sampling and metaheuristics, and concluded that simulated annealing procedure of Bouleimen and Lecocq [48] and the genetic algorithm of Hartmann [49], belonging to metaheuristics, were most successful [50]. Avella, Boccia, and Sforza [51] developed a penalty function heuristics for RCSPP based on the extension to the discrete case of an exponential penalty function heuristic proposed for the solution of large-scale LP's. Merkle, Middendorf, and Schmeck [52] developed an ant colony optimization algorithm for RCPSP and showed the advantages over several other heuristics, including genetic algorithm, simulated annealing, tabu search, and different sampling methods. Our addressed problem in this chapter, although belonging to *RCSP*, is not a typical *RCPSP* or *RCSPP*. The problem needs more study due to its own cost and constraint structure.

4.2 Algorithm Development and Analysis

Based on the algorithm developed for the problem in Chapter 3 where lead time constraint is not considered, a revised DP algorithm will be introduced, summarized, and analyzed in this section.

4.2.1 DP Algorithm Description

For ease of algorithm description, we give the following definitions. Define $d_i = |\mathbf{A}_i|$, i.e., d_i is the number of feasible ways to build subassembly v_i. Define $\mathbf{b}_i$ as the array of indices of AND relations belonging to subassembly v_i, i.e., $\mathbf{b}_i(k) = j$ where $leb(v_p^j) = i + N$, $j \in \mathbf{I}_a$.

Define (i,k,s,t) as the state, where i is the label of CV, i.e., $i = leb(v)$ ($N+1 \le i \le N+M$); k is the index of AND relation of vertex v ($k \le d_i$); s is the supplier of the assembly process, where $s \in \mathbf{Q}_l$ and $l = fapi(\mathbf{b}_i(k))$; t ($t \le T(i)$) is the lead time for producing subassembly v_i, where $T(i)$ is an upper bound for the lead time of subassembly v_i.

To ensure the lead time of the final product is within lead-time limit T, the subassemblies in the lower levels should have lower lead-time upper bounds than those in the higher levels. Minimizing the computational work is usually achieved by estimating proper upper bounds for intermediate subassemblies. However, this requires solving another optimization problem: finding the optimal upper bounds. Since this dissertation focuses on providing one solution to the time-constrained optimization problem, we loosen the upper bound for the lead times of the intermediate subassemblies. Define $T(i)=\begin{cases} T-1, i<N+M \\ T, i=N+M \end{cases}$ so that all solutions at the intermediate levels with a lead time less than T will be considered as part of the possible optimal solutions. Hence, we consider solving the optimization problem for each (i,k,s) at lead-time levels from 1 to $T-1$, where $i<N+M$.

Define the value function $V(i,k,s,t)$ as the minimum supply chain cost to produce subassembly v_i within lead time t, given assembly choices (represented by AND relation index k) and the associated process provider s ($s \in \mathbf{S}_l$ and $l = fapi(\mathbf{b}_i(k))$). By setting a time bound t, we guarantee that $V(i,k,s,t_1) > V(i,k,s,t_2)$ if $t_1 < t_2$, which means that when $t_1 < t_2$, if $V(i,k,s,t_1)$ is less than or equal to $V(i,k,s,t_2)$, the latter will not be an optimal solution.

Then the task of each subassembly supplier is to minimize its supply chain cost by selecting how to assembly the subassembly and where to acquire its two immediate child subassemblies. At state (i,k,s,t), supplier s has two subassemblies, $v_l^j = \mathbf{a}_j(2)$ and $v_r^j = \mathbf{a}_j(3)$, where $j = \mathbf{b}_i(k)$. The decision process is described below.

For ease of description, we denote the index of vertex v_l^j as i_1, i.e., $i_1 = leb(v_l^j)$. Vertex i_1 has d_{i_1} AND relations with the index array $\mathbf{b}_{i_1}$. For each assembly method k_1 ($k_1 \le d_{i_1}$), the corresponding assembly process to accomplish is $l_1 = fapi(j_1)$, where $j_1 = \mathbf{b}_{i_1}(k_1)$. Hence we may select the supplier $s_1 \in \mathbf{S}_{l_1}$ for subassembly i_1. The costs related to subassembly i_1 include two parts: 1) procurement cost $V(i_1,k_1,s_1,t_1)$ of purchasing i_1 from supplier s_1 within lead time t_1; 2) transportation cost $tc(i_1,s,s_1)$ from supplier s_1 to supplier s. The lead time related to subassembly i_1 consists of the lead time t_1 of the subassembly itself and the transportation time $tt(i_1,s,s_1)$ between two suppliers.

Similarly, vertex v_r^j is denoted as i_2. For each assembly method k_2 ($k_2 \le d_{i_2}$), the corresponding assembly process to accomplish is denoted as $l_2 = fapi(j_2)$, where $j_2 = \mathbf{b}_{i_2}(k_2)$. The costs related to subassembly i_2 include procurement cost $V(i_2,k_2,s_2,t_2)$ and transportation cost $tc(i_2,s,s_2)$. The lead time related to subassembly i_2 consists of the lead time t_2 and the transportation time $tt(i_2,s,s_2)$.

Then the lead time of v_i is $lt = \max\{t_1 + tt(i_1,s,s_1), t_2 + tt(i_2,s,s_2)\} + at(i,s)$. Thus, the value function is as follows:

$$V(i,k,s,t) = \min_{k_1,s_1,t_1;k_2,s_2,t_2 | lt \le t} (V(i_1,k_1,s_1,t_1) + tc(i_1,s,s_1) + V(i_2,k_2,s_2,t_2) + tc(i_2,s,s_2)) + ac(i,s)$$

where $i_1 = leb(\mathbf{a}_i(2))$, $k_1 \le d_{i_1}$, $l_1 = fapi(\mathbf{b}_{i_1}(k_1))$, $s_1 \in \mathbf{S}_{l_1}$, $t < T(i_1)$;

$i_2 = leb(\mathbf{a}_i(3))$, $k_2 \le d_{i_2}$, $l_2 = fapi(\mathbf{b}_{i_2}(k_2))$ $s_2 \in \mathbf{S}_{l_2}$, $t < T(i_2)$;

$k \le d_i$, $l = fapi(\mathbf{b}_i(k))$, $s \in S_l$, $t < T(i)$.

The minima will be recorded to trace back the assembly plans:

$$opt(i,k,s,t) = (k_1^*, s_1^*, t_1^*, k_2^*, s_2^*, t_2^*).$$

While the CVs in the HAG are linked to deciding assembly plans and relevant suppliers, the only decision an SV is associated with is supplier assignment (from whom to purchase components). Therefore, the value function at an SV is the component purchasing cost:

$$V(i,k,s,t) = pc(i,s),\ k = 1,\ s \in \mathbf{S}_i,\ \text{if } i \le N.$$

Depending on business scenarios, the final step may vary. In the scenario where the objective is to optimize the manufacturing cost of products at one specific final assembler, the final step is to pinpoint the optimal solutions among d_{M+N} assembly choices at the final vertex. Define V as the optimal cost at the final assembler, then $V = \min_{k,t|t \le T} V(N+M,k,s,t)$, where $s = 1$. In another scenario where the objective is to optimize the cost of products to markets, the final assembler is also a choice to make when multiple final assemblers exist. Define V as the optimal cost to the market, then $V = \min_{k,s,t|t \le T} (V(N+M,k,s,t) + tc(N+M,s,s_0))$, where s_0 is the location of the market.

4.2.2 DP Formulation Summary

The summary of the DP formulation of the algorithm is as follows:

State: (i,k,s,t) , where i is the index of compound vertex v , i.e. $i = leb(v)$, $N+1 \le i \le N+M$; k is the AND relation index of vertex v ($k \le d_i$); s is the supplier of assembly process, where $s \in \mathbf{Q}_l$, $l = fapi(j)$, and $j = \mathbf{b}_i(k)$; t is the lead time for producing subassembly v_i .

Value Function: The minimum total supply chain cost to produce compound vertex i with lead time t at most is denoted as $V(i,k,s,t)$, given the assembly type, i.e., the AND relation index $k \le d_i$, as well as the corresponding manufacturing process provider $s \in S_l$ for process l , where $l = fapi(j)$.

Functional Equation:

$$V(i,k,s,t) = \min_{k_1,s_1,k_2,s_2 | lt \le t} (V(j_1,k_1,s_1,t_1) + tc(s,s_1) + V(j_2,k_2,s_2,t_2) + tc(s,s_2)) + ac(i,s) \text{, where}$$

$$lt = \max\{t_1 + tt(s,s_1), t_2 + tt(s,s_2)\} + at(i,s)$$

Boundary Condition: $V(i,k,s,t) = pc(i,s), k = 1, s \in \mathbf{S}_i$.

The flow char for the algorithm is shown in Figure 4.1.

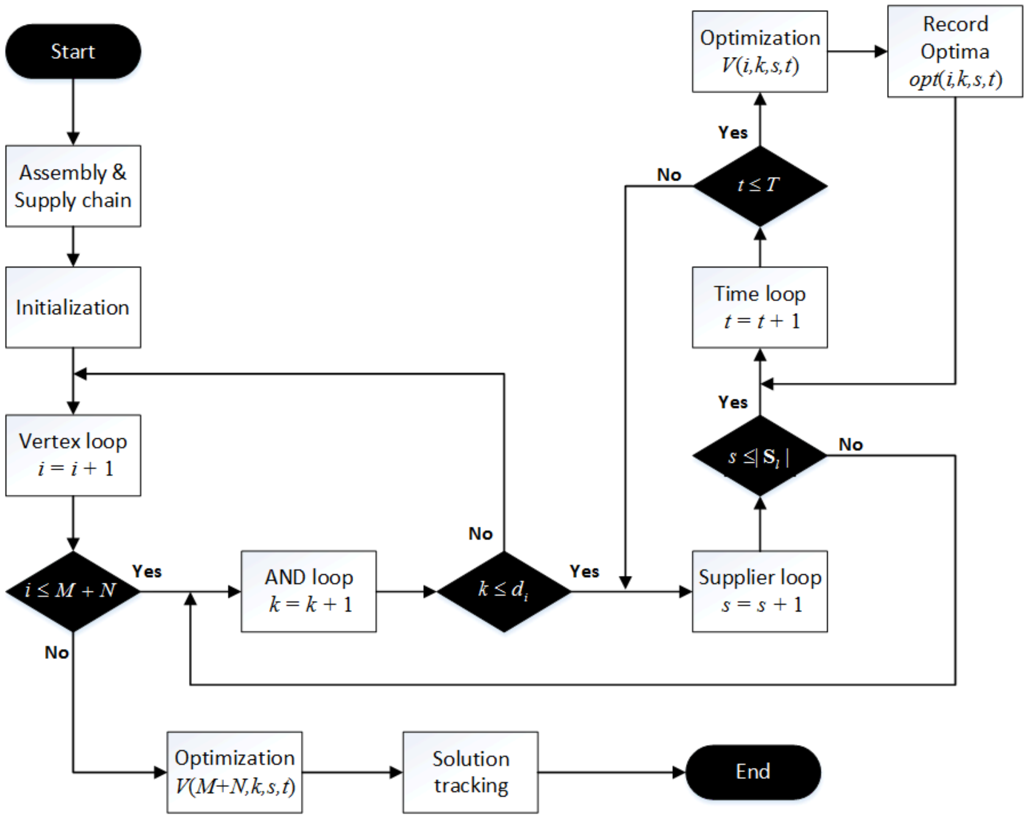


Figure 4.1. DP algorithm flow chart

4.2.3 Computational Complexity Analysis

A similar analysis can be found in the previous chapter. A upper bound for the computational complexity of the algorithm is $O(MD^2S^2T^2)$, where M is the number of CV in HAG, D is the maximum number of possible ways to assemble a subassembly, S is the maximum number of suppliers for a component or a manufacturing process, T is the time boundary. In typical products, $M = O(\mathrm{N}^2)$ and $\mathrm{D} = O(\mathrm{N})$, and the computational complexity is $O(N^4S^2T^2)$. In contrast, the enumeration has a computational complexity of $O(\mathrm{KS}^{L+N})$, where K is the number of assembly plans, L is the number of processes required, N is the number of components in the product, and S is the maximum number of suppliers for a component or a process. By taking advantage of the compactness of proposed Hyper AND/OR graph and searching in a structured way, the DP algorithm transforms the problem with exponential time complexity to one with polynomial time complexity. More research in upper bounds of lead time, *T(i)*, can offer more reduction space of the computational complexity. The $O(MD^2S^2T^2)$ complexity does not contradict the fact that the integrated problem is NP-hard, since T is proportional to the number of bits in $\log(N)$ not N theoretically. However, in real cases, T is usually within a considerate range, the algorithm could solve most cases within a reasonable time range.

4.3 Laptop Computer Case Study

The laptop case in Chapter 3 is applied to evaluate the performances of the developed algorithm. District 1 and 3 can provide all the assembly processes, while District 2 can offer some of the assembly work related to display. 52 suppliers, providers of components or assembly processes, exist in the supply chain. If two suppliers are in the same district, the transportation cost is low, roughly \$1 and the transportation time is 3 day. Otherwise if they are companies located in different districts of the first three districts, the transportation cost is high, around \$6 and the transportation time is 20 days. If the transportation is between District 4 and the other districts, the cost is the highest, around \$10, and the transportation time is 30 days.

If the time bound is set to be 40, the optimal assembly and supply chain plan generated by the proposed method is shown in Figure 4.3.

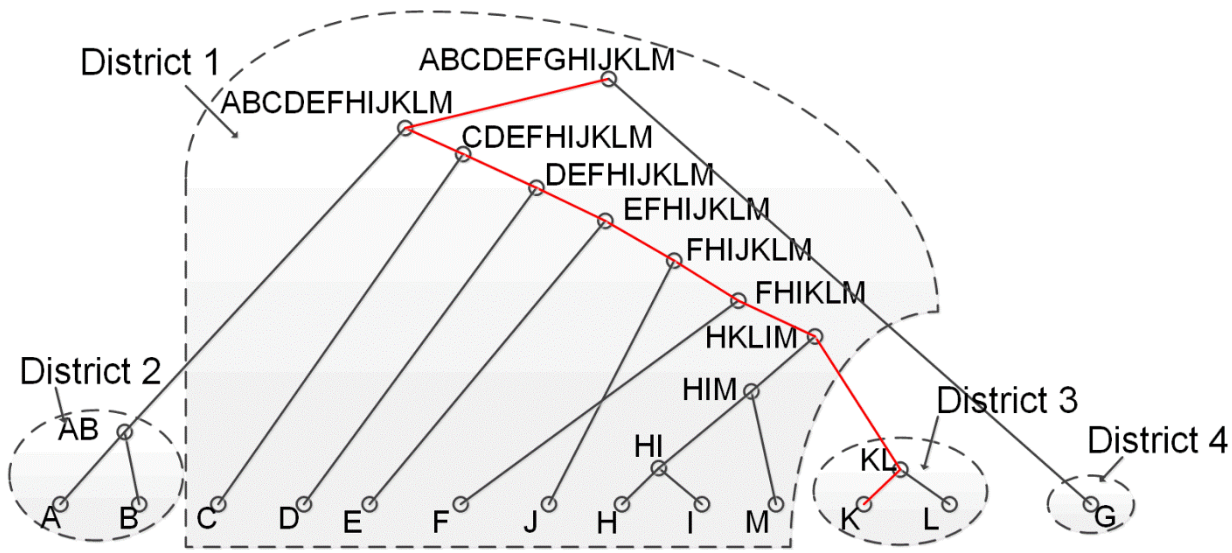


Figure 4.2. The optimal assembly plan with supplier assignment, lead time 39 days.

Define a route from a SV to the final vertex a path and the number of vertices along one path as its length. And define the length of the longest path the depth of the assembly plan. The longest path is (H,HI,HIM,HKLIM,...,ABCDEFGHIJKLM) with length 9. The optimal assembly plan has depth of 9. Label the path with the longest lead time in red. Due to the long transportation time between district 3 and district 1, path in red has the longest lead time although it is not the longest path. The result is almost the same as that in the previous chapter as shown in Figure 4.3, except that keyboard & palm rest are assembled into one subassembly. For ease of description, call the optimal solution with no time constraint Plan I, and the one with lead time constraint 40 days Plan II. Two plans have the same optimal cost, since they have the same supplier assignment. However, two plans differentiate on depth, while Plan I has a depth of 11, Plan II has a depth of 9. Plan II can potentially shorten its lead time through decreasing the depth of the plan. In other words, the proposed method could balance production of multiple subassemblies to shorten the lead time.

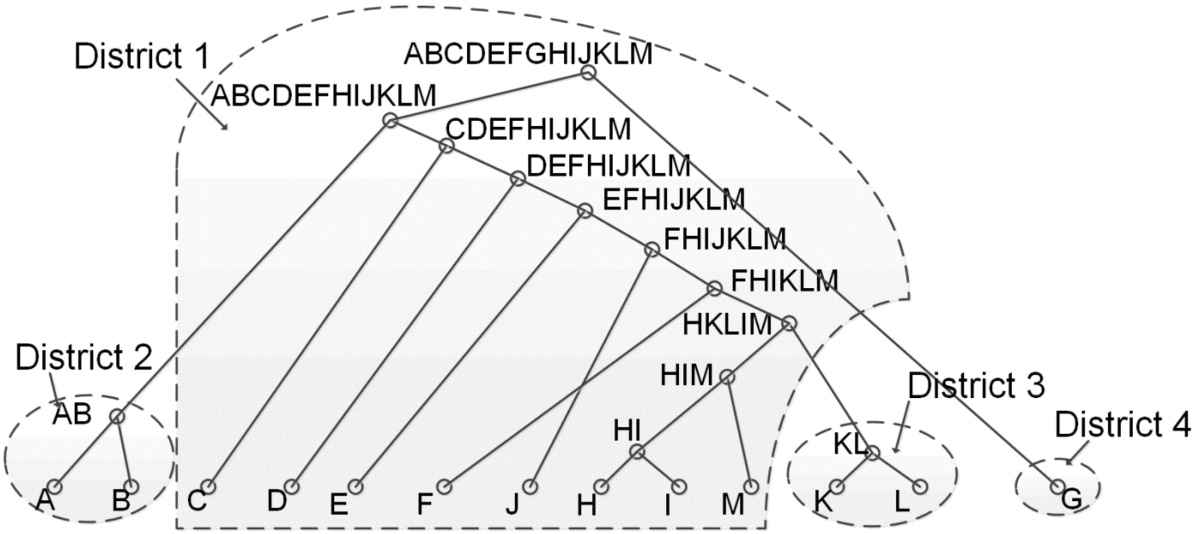


Figure 4.3. The optimal configuration, no lead time constraint.

If time bound decreases from 40 to 35, the display subassembly cannot be assigned to district 2 anymore. The optimal solution is to get processor subassembly in district 1, although the total supply chain cost increases by a couple of dollars. The optimal solution with 34 days as time bound could be found in Figure 4.4. And the path with longest lead time switches to the longest path.

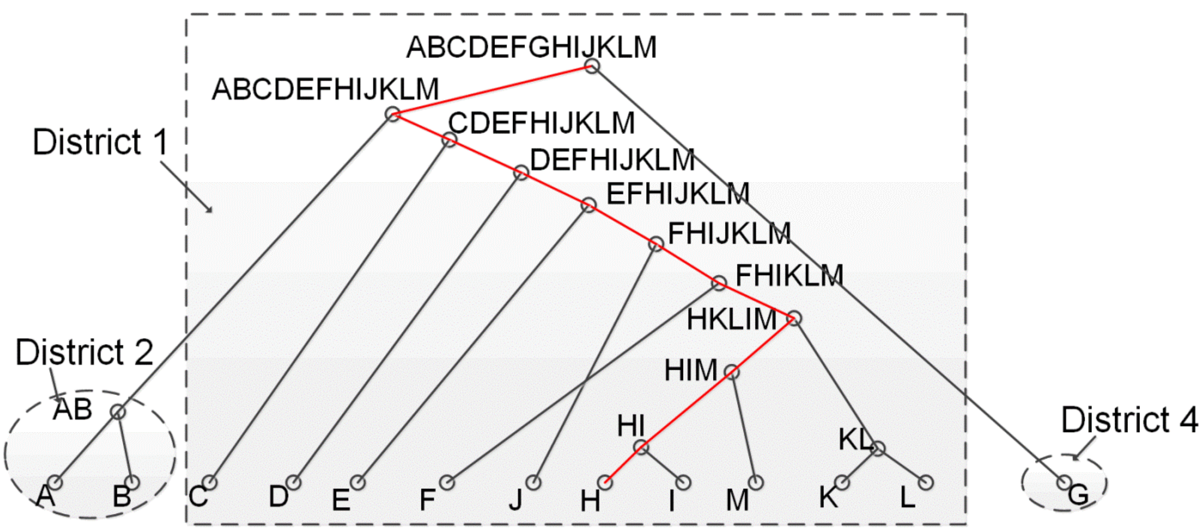


Figure 4.4. The optimal assembly plan with supplier assignment, lead time 34 days

As shown above, this model can efficiently make integrated assembly planning and supplier assignment decisions for various products. The result also demonstrates its potential to balance parallel subassembly productions, which could shorten lead times. The balance here refers to height-balance of the assembly trees in the HAG, which means that a more balanced assembly plan tends to have lower height. In this dissertation, we call the height of the assembly plan as its length.

The DP algorithm is run on a regular laptop using MATLAB, it only took 1.8 secs to obtain the optimal solutions. An enumerative algorithm is applied to find the optimal assembly plan and supplier assignment among 5280 assembly plans and around 3.5 million supply chain configurations for each assembly plan. Although the running time for one iteration, i.e. each assembly plan and one of its supply chain configurations, is only 0.028 secs, the total running time is $0.028 \times 5280 \times 3500000 \approx 5.2 \times 10^{8}$ secs, i.e. 16.4 years. This shows that our algorithm could sharply decrease the computational time through structural search.

To further validate the computational efficiency of the proposed method, we ran the algorithm on randomly generated assemblies of small to medium sizes. These assembly structures, containing 5 to 24 components, were created to simulate realistic products ranging from office supplies to medium size electronics. Each component or assembly process was assumed to have three candidate suppliers. We ran the algorithm 8 times for each assembly size and calculated the average running time and size of hyper AND/OR graphs, considering that the randomness of the simulated product structures and task precedence affects the decision spaces. The average value of the experiment data is plotted against the size of assemblies using double log coordinates, as shown in Figure 4.5 and 4.6.

The former shows the HAG size, and the latter shows the computation time. Here, the circles represent the average values and the line represents the best linear fitting. The linear fitting between the logarithmic coordinates show the orders of the size of HAG and the running time with respect to the size of the assemblies. The results show the HAG size is in the second order of assembly size and running time is in the fourth order. These numbers are consistent with our theoretical analysis.

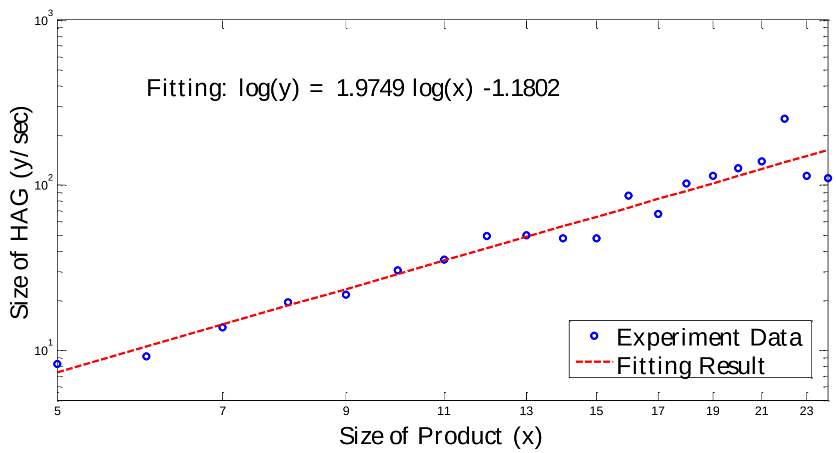


Figure 4.5. Size of HAG vs. size of product

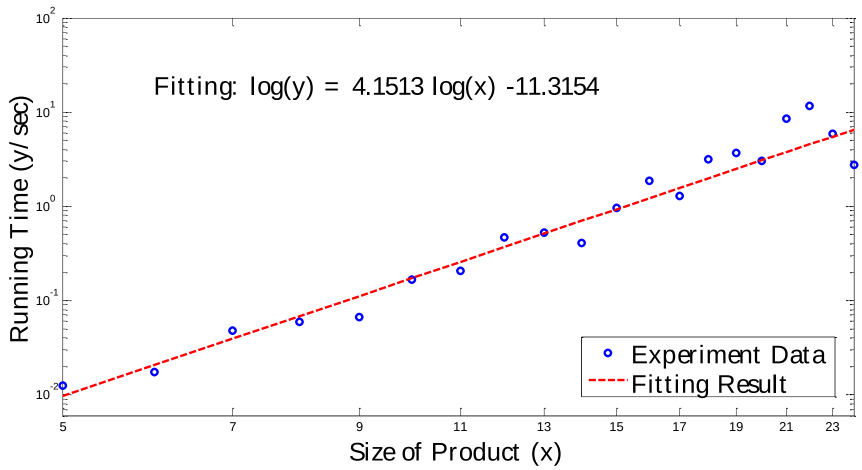


Figure 4.6. Running time vs. size of product

4.4 Summary

This chapter presents a systematic analysis of the time constrained problem of integrated design of assembly and supply chain through problem identification, feature demonstration, and solution discussion. A bottom-up DP algorithm with pseudo-polynomial time complexity is developed to search for the global optimal assembly plan and supply chain configuration with the lowest total cost under time constraint. The method has been verified through a case study of a laptop computer assembly.

Although the upper bound of the lead time is loosened in each subassembly levels, the DP algorithm still solves the integrated decision problem with exponential time complexity using pseudo-polynomial computational time. More research in optimal upper bounds can help further reduce the computational complexity of the solution algorithms. The developed model can be extended to address other constrained optimization problems such as indoor material handling and transportation planning between facilities.

Applicability of the developed method may be demonstrated by the experiences in several industries in the past decades. For instance, Doran et al. surveyed suppliers at different tiers of French automobile manufacturers and identified how the shift from traditional to modular manufacturing and supply chain influenced the operations of key suppliers [33]. The results showed that modular approach was beneficial in many aspects such as cost, delivery time, and development time. Thus, the key suppliers attempted to transfer low value-adding activities (such as some plastic injection molding) to lower-tier suppliers while they were focusing on a few core businesses. However, contrary to the common expectation, the modular manufacturing and supply chain sometimes brought adverse effects. Accommodating such modular approaches might increase the management complexity in terms of module design, module selection, and quality control. The modular approach also required a new buyer-supplier relationship. The larger modules produced in the suppliers may lead to more profits in the suppliers but less profits in the main assemblers due to the increased module prices. Therefore, a better understanding on the interdependency of supplier selection and assembly plans for concurrent decisions is desirable so as to find better modular strategies.

If the method is developed further and combined with other enterprise information technologies, it has a potential for strong practical impact in industry. The developed method in this chapter can be utilized to decide which activities to be retained at the assemblers themselves and which to be transferred to lower-tier suppliers, in particular in module design. When linked with powerful enterprise database, our method can generate optimal or quasi-optimal solutions for decision makers in each tier of the supply chain. This will help a supply chain to gain more competitiveness in the market.

CHAPTER 5

Assembly Supply Chain Design for a Product Family

5.1 Introduction

Manufacturing companies are endeavoring to provide high variety of products to demanding and heterogeneous customers in order to strengthen their competitiveness in the global market. The proliferation of products places pressure on manufacturers as inventory control of components and subassemblies is becoming more and more complicated. For example, General Motors (GM) carried 131 different rear axle assemblies in its pickup truck division [53]. The uncertainty in customer demands even makes the inventory control more challenging. Safety stock (also called buffer stock) is a term used to describe a level of extra stock that is maintained to mitigate risk of stock outs due to uncertainties in supply and demand [54]. Serving as an insurance against stock outs, safety stock is held when there is uncertainty in demand, supply, or manufacturing yield. Adequate safety stock levels permit business operations to proceed according to their plans.

While hedging the supply chain risks through safety inventory, manufacturers are facing the pressure to store excessive inventory in their supply chains. Risk pooling is an important concept in supply chain management for higher efficiency [55]. It suggests that demand variability is reduced if demand is aggregated because it becomes more likely that high demand from one customer will be offset by low demand from another when demand is aggregated. This reduction in variability allows a decrease in safety stock and thus reduces average inventory. Risk pooling may appear in various forms, such as early design collaborations, sharing product rollover plans, and supplier hubs [56].

To take full advantages of risk pooling, manufacturing firms are seeking opportunities to build efficient assembly supply chains with efforts in product development, assembly planning, and supply chain management. The efficiency of assembly supply chains in this dissertation is defined as the capability to avoid wasting materials, capital, and time in producing a variety of products to meet customers' demands. Among all the opportunities to increase assembly supply chain efficiency, a proper design of assembly plans serves as the main contributor due to its significant impact on supply chain management, production planning, and scheduling. Research has been conducted on coping with the variety induced complexity and uncertainty in assembly planning and supply chain management in order to enhance the efficiency of assembly supply chains. The efforts can be generally divided into two categories: index-based optimization and heuristic strategies. Index-based optimization develops new indices to represent the efficiency of assembly and supply chain systems, assuming that optimizing the indices will yield optimal system performances. The heuristic strategy studies focus more on discovering rules and strategies to guide assembly supply chain designs although they also optimizes on common indices such as average inventory levels, mean customer waiting time, and product costs. A brief review of the literature in both categories is presented below.

In the research of the first category, minimizing complexities of assembly supply chains is equated to improving their efficiency. Information entropy is an intuitive measure for system complexity as entropy is defined as a measure of uncertainty. Deshmukh et al.

[57] defined "static complexity" as a function of the structure of the system, the variety of sub-systems, and strength of interactions, and correlated the concept with average waiting time in the manufacturing systems. Hu et al. [58] utilized the uncertainty in product mix to represent the source of complexity and defined station and system level complexity. An algorithm was developed to mitigate the complexity. Kuzgunkaya and ElMaraghy [59] presented a new metric accounting for the complexity inherent in the various modules in the manufacturing system through the use of an index derived from a newly developed manufacturing systems classification code. Wang and Hu [60] proposed a measure of product variety induced manufacturing complexity based on the choices of assembly activities that operators make in serial, manual mixed-model assembly lines. Other indices are mostly graph based. Ishi and Martin [61] proposed "Process Sequence Graph" and they equated the node-count reduction to a reduction in inventory and complexity costs. Modrak and Marton [62] applied a vertex degree index for measuring a structural complexity of assembly supply chain networks. Although these studies proposed creative indices and provided propotential insights into efficient assembly supply chain designs, they usually did not establish explicit connections between the proposed indices and common measures such as costs, time, and quality. The outcomes of the research hence are difficult to be verified and applicable in real industrial settings.

The research in the heuristic strategy is more successful in generating realistic and applicable results as it focuses mainly on business measures, such as average inventory level and lead time. One of the most widely accepted strategies is delayed differentiation. Also called postponement, delayed differentiation is a concept in supply chain management where the manufacturing process starts by making a generic product that is later differentiated into specific end-products. This widely used method can be effectively adopted to address the final demand even if forecasts cannot be accurate enough, especially in industries with high demand uncertainty. Research has been conducted from various facets of delayed differentiation as reviewed in [63 - 66]. Lee and Tang [67] built a mathematic model that captures the costs and benefits related to the redesign strategy for postponement. The optimal point of product differentiation was characterized and managerial insights were derived. Swaminathan and Tayur [68] modeled the problem of finding semi-finished products (vanilla boxes) as a two-stage integer program with recourse and provided an effective solution procedure by utilizing structural decomposition of the problem and (sub) gradient derivative methods. Gupta and Krishnan [69] formalized the notion of generic subassemblies and presented an algorithm to identify the generic subassemblies so as to maximize the benefits from commonality of components and assembly operations, referred to as product family-based assembly sequence design. Forza, Salvador, and Trentin [70] defined three mutually exclusive and exhaustive types of form postponement (FP) at the company level of analysis and formalized how, why, and under which assumptions each FP type affects operational performance. The strategies have been acknowledged in academia and industry due to its simplicity for applications.

The central idea of delayed differentiation is essentially risk pooling through delaying uncertainty to later stages. An implicit assumption lies underneath the strategies that common processes exist throughout product families. However, the commonality may be very limited for some products or companies in today's manufacturing. Blecker and Abdelka [71] presented the insufficiencies of the delayed product differentiation principle by means of a simple example from the computer industry in which the degree of product modularity is very high. They demonstrated that this principle cannot support optimal decisions concerning how variety should proliferate throughout the assembly process. These naturally lead to a question: what is the optimal strategy when commonality is limited or more generally what is the room for further optimization when we have varying levels of commonality? This chapter aims at answering this question. Specifically, this chapter presents a mathematical model with the objective to minimize inventory considering a product with multiple components and processes that could be differentiated. Starting from a discussion on a two process product, a theorem about the impact of product variety on their safety inventory is developed. Then a measure to approximate the impact is derived. Two types of problems, sequencing and decomposition, will be investigated and one case study will be provided.

This chapter is organized as follows: Section 2 builds a mathematic model and develops theorems concerning the impact of variety on inventory in a general product family. Section 3 focuses on the process sequencing problem. Section 4 covers the optimal assembly decomposition problem, including a model and algorithm. Section 5 illustrates the concept and the applications of the developed method. Section 6 concludes the chapter.

5.2 Impact of Variety on Inventory

Consider a product with multiple components requiring multiple processes to assemble the components into the final product. In each step, multiple components or subassemblies are assembled into one larger subassembly or finally into the final product. In addition to the assembly process, the subassembly may require extra manufacturing processes to finish the subassembly, e.g., finishing, polishing, and cutting. A subassembly may need multiple processes with different numbers of variants. For instance, cutting process may have different shapes and polishing may have different surface requirements. How to arrange the sequence of the manufacturing processes so as to mitigate the variety induced uncertainty is defined as a process sequencing problem. On the other hand, different assembly plans represent different combinations of components or subassemblies into higher level subassemblies. Different combinations may also generate different inventory types or levels. Which combination/assembly plan carries the lowest inventory level is what we refer to as an assembly decomposition problem. Before addressing the two problems directly, we will investigate first how the variety of a subassembly affects its inventory. We will start with a simple two process example. Then the problem will be formalized and a discussion will be extended to more general cases.

5.2.1 Introductory Example

Consider producing bicycle handles with one compass and grips in two colors (black and red) as shown in the figures below. A pair of grips and one compass need to be assembled to finish the handle assembly. For ease of description, we will denote the subassemblies by symbols and acronyms. We will denote the handles without grips or compass assembled the basic handles (BH).

Figure 5.1. Fully assembled bicycle handles and components

Assume the demand of the handles with black grips and compass follows a normal distribution of (μ_1, σ_1) and that of the handles with red grips and compass follows a different normal distribution of (μ_2, σ_2). Assume these two demands are independent from each other. Assume the service level is z. If grips are assembled first as shown in Figure 5.2, the average inventory level of the handles with black and red grips assembled will be $\frac{\mu_1}{2} + z\sigma_1$ and $\frac{\mu_2}{2} + z\sigma_2$. The total inventory of the handles with grips will be $\frac{\mu_1+\mu_2}{2} + z(\sigma_1 + \sigma_2)$.

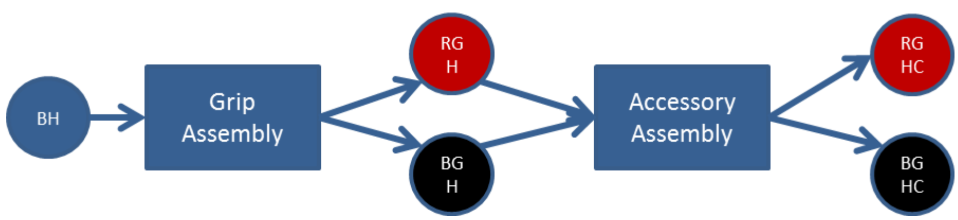


Figure 5.2. Grip assembly first sequence

However if the compass is assembled first, the average inventory level for the handles with compasses assembled is $\frac{\mu_1+\mu_2}{2} + z(\sqrt{\sigma_1^2 + \sigma_2^2})$. Obviously $(\sigma_1 + \sigma_2) \geq \sqrt{\sigma_1^2 + \sigma_2^2}$ since σ_1 and σ_2 are non-negative. The reduction of average inventory due to demand pooling is regarded as one of the main benefits of delayed differentiation.

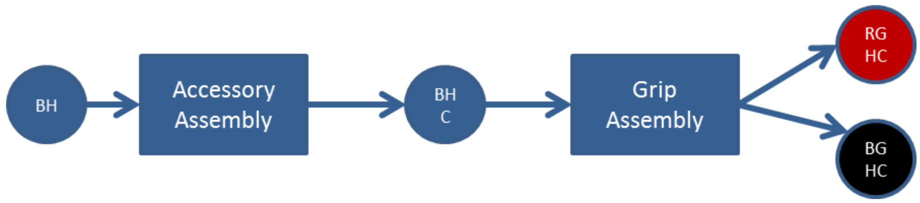


Figure 5.3. Compass assembly first sequence

A direct extension of the problem is what if there are no common processes, e.g., when there are multiple accessories, say compass, ring, and light. The example product mix is shown in the table below, assuming product variant demands are independent from each other.

Table 5.1. Product mix

Mix	**Compass**	**Ring**	**Light**
Red	(μ_{11}, σ_{11})	(μ_{12}, σ_{12})	(μ_{13}, σ_{13})
Black	(μ_{21}, σ_{21})	(μ_{21}, σ_{22})	(μ_{23}, σ_{23})

The illustration exhibits can be found in figures below, where blocks represent buffers and the lines between blocks represent the different processes.

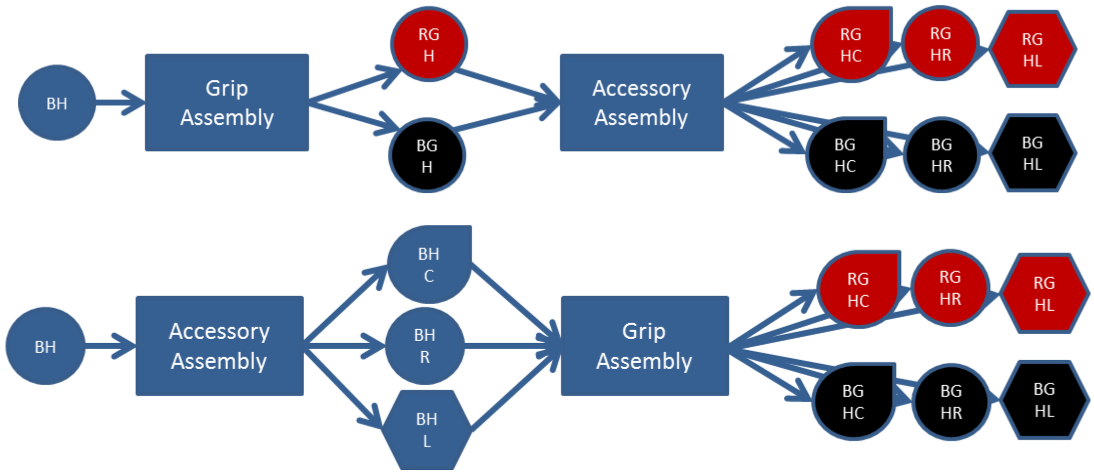


Figure 5.4. Two assembly sequences for bicycle handles

The inventory levels can be calculated for each buffer. The inventory levels for the end products and the basic handles are definitely the same for the two sequences. Thus the only difference lies in the intermediate inventories. For sequence 1, the inventory level for the handles with red and black grips are respectively $\frac{\mu_{11}+\mu_{12}+\mu_{13}}{2}+z\sqrt{\sigma_{11}^2+\sigma_{12}^2+\sigma_{13}^2}$, $\frac{\mu_{11}+\mu_{12}+\mu_{13}}{2}+z\sqrt{\sigma_{21}^2+\sigma_{22}^2+\sigma_{23}^2}$. Hence the total intermediate inventory level can be calculated as $\sum_{i,j}\frac{\mu_{ij}}{2}+z(\sqrt{\sigma_{11}^2+\sigma_{12}^2+\sigma_{13}^2}+\sqrt{\sigma_{21}^2+\sigma_{22}^2+\sigma_{23}^2})$. For sequence 2, the inventory levels for handles with compass, ring, and light are $\frac{\mu_{11}+\mu_{21}}{2}+z\sqrt{\sigma_{11}^2+\sigma_{21}^2}$, $\frac{\mu_{12}+\mu_{22}}{2}+z\sqrt{\sigma_{12}^2+\sigma_{22}^2}$, and $\frac{\mu_{13}+\mu_{23}}{2}+z\sqrt{\sigma_{13}^2+\sigma_{23}^2}$. The total intermediate inventory level is $\sum_{i,j}\frac{\mu_{ij}}{2}+z(\sqrt{\sigma_{11}^2+\sigma_{21}^2}+\sqrt{\sigma_{12}^2+\sigma_{22}^2}+\sqrt{\sigma_{13}^2+\sigma_{23}^2})$. Denote $\sqrt{\sigma_{11}^2+\sigma_{12}^2+\sigma_{13}^2}+\sqrt{\sigma_{21}^2+\sigma_{22}^2+\sigma_{23}^2}$ as D_1 and $\sqrt{\sigma_{11}^2+\sigma_{21}^2}+\sqrt{\sigma_{12}^2+\sigma_{22}^2}+\sqrt{\sigma_{13}^2+\sigma_{23}^2}$ as D_2. Then the difference of total inventory levels between sequences 1 and 2 is $z(D_1-D_2)$, which is only related to their uncertainty part (i.e., variation). It shows that the difference of safety inventory between the subassemblies only with grips and only with accessories determines the performance of the two assembly sequences. As we can see, the key to the problem is to identify the relationship between subassembly variety and its safety inventory.

We will provide a mathematical representation of the example problem and an answer in the form of lemma in the following section.

5.2.2 In-depth Analysis of the Example Problem

Define M as the matrix of the standard deviation of product demand, i.e., $M = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \end{pmatrix}$. Then D_1 and D_2 can be expressed as the L2,1 norm of matrix M^T and M, where L2,1 norm [72] is the sum of the Euclidean norm of the columns of the matrix. L2,1 norm of a standard deviation matrix hence represents the uncertainty level of the inventory of specific subassemblies. $L2{,}1(M^T)$ represents the uncertainty level of the handles with only grips while $L2{,}1(M)$ represents that of the handles with only accessories. The uncertainty level difference therefore is equivalent to the difference between $L2{,}1(M^T)$and $L2{,}1(M)$.

From the discussion above, we can see that L2,1 norm is an asymmetrical operator on matrices as L2,1 norm of a matrix is not necessarily equal to that of its transpose. Our objective is to study the asymmetry of L2,1 norm and how the asymmetry is related to the matrices. Numerical studies show that there is no deterministic relationship between L2,1 norm of M and that of its transpose; $L2{,}1(M^T)$ can be greater/less than or equal to $L2{,}1(M)$ depending on M. To nullify the dependence on M so as to unveil the structural difference between these two norms, we assume the standard deviation ($\sigma_{i,j}$) of each product is unknown and follows i.i.d. distributions. Multiple types of distributions may be assumed for the deviations, e.g., normal, uniform, and exponential. We will assume normal distribution, specifically with zero mean, to conduct theoretical studies in this dissertation. Numerical studies provided later will show that different distribution assumptions actually would not change the analytical results. Also considering $\sigma_{i,j}$ non-negative, we assume $\sigma_{i,j} = \bar{\sigma}|z_{i,j}|$, where $\bar{\sigma}$ is a coefficient and $z_{i,j}$ follows standard normal distribution. To ensure that the probability density function of $\sigma_{i,j}$ is continuous at $\sigma = 0$, define $f(\sigma = 0) = \frac{2}{\sqrt{2\pi\bar{\sigma}}}$.

We can now provide a formal problem statement. Given a $m \times n$ matrix M, $L2{,}1(M^T) = \sum_{i=1}^{m} \sqrt{\sum_{j=1}^{n} \sigma_{ij}^2} = \bar{\sigma} \sum_{i=1}^{m} \sqrt{\sum_{j=1}^{n} z_{ij}^2}$ and $L2{,}1(M) = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{m} \sigma_{ij}^2} = \bar{\sigma} \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{m} z_{ij}^2}$, where z_{ij} follows i.i.d. standard normal distribution, our objective is to discover if there is certain inequality between $\mathbf{E}(L2{,}1(M^T))$ and $\mathbf{E}(L2{,}1(M))$ when $m < n$.

Denote $Y(t) = \sqrt{\sum_{i=1}^{t} z_i^2}$. Then $\mathbf{E}(L2{,}1(M^T)) = \bar{\sigma} m \mathbf{E}(Y(n))$ and $\mathbf{E}(L2{,}1(M)) = \bar{\sigma} n \mathbf{E}(Y(m))$. Obviously $Y(n)$ follows a Chi distribution with t degrees of freedom, $\mathbf{E}\big(Y(t)\big) = \sqrt{2}\frac{\Gamma(\frac{t+1}{2})}{\Gamma(\frac{t}{2})}$, and $\mathbf{Var}\big(Y(t)\big) = t$.

If t is even, i.e., $t = 2k$, then $\mathbf{E}\big(Y(t)\big) = k\sqrt{2\pi}\frac{(2k)!}{k!k!4^k}$. If t is odd, i.e., $t = 2k + 1$, then $\mathbf{E}\big(Y(t)\big) = \sqrt{\frac{2}{\pi}\frac{k!k!4^k}{(2k)!}}$. We will discuss the inequality between $\mathbf{E}(L2{,}1(M^T))$ and $\mathbf{E}(L2{,}1(M))$ in four scenarios, i.e., when m and n are even or odd.

<u>Scenario 1</u><u>: $m = 2l < n = 2k$</u>

$\mathbf{E}(L2{,}1(M^T)) = m\mathbf{E}\big(Y(n)\big) = 2lk\sqrt{2\pi}\frac{(2k)!}{k!k!4^k}$ and $\mathbf{E}(L2{,}1(M)) = n\mathbf{E}\big(Y(m)\big) = 2kl\sqrt{2\pi}\frac{(2l)!}{l!l!4^l}$. $\rightarrow \frac{\mathbf{E}(L2,1(M^T))}{\mathbf{E}(L2,1(M))} = \frac{(2k-1)!!(2l)!!}{(2l-1)!!(2k)!!} = \frac{(2k-1)(2k-3)\dots(2l+1)}{(2k)(2k-2)\dots(2l+2)} < 1.$

Hence $\mathbf{E}(L2{,}1(M^T)) < \mathbf{E}(L2{,}1(M))$.

<u>**Scenario 2**: $m = 2l - 1 < n = 2k, l \le k$</u>

$\mathbf{E}(L2{,}1(M^T)) = m\mathbf{E}\big(Y(n)\big) = (2l-1)k\sqrt{2\pi}\frac{(2k)!}{k!k!4^k}$,

$\mathbf{E}(L2{,}1(M)) = n\mathbf{E}\big(Y(m)\big) = 2k\sqrt{\frac{2}{\pi}}\frac{(l-1)!(l-1)!4^{l-1}}{(2(l-1))!} = 2k\sqrt{\frac{2}{\pi}}\frac{l!l!4^l(2l-1)}{(2l)!(2l)}$.

$\rightarrow \frac{\mathbf{E}(L2,1(M^T))}{\mathbf{E}(L2,1(M))} = \pi l\frac{(2k-1)!!(2l-1)!!}{(2k)!!(2l)!!} \le \pi l\left(\frac{(2l-1)!!}{(2l)!!}\right)^2 < 1$. The first inequality is due to that $f(t) = \frac{(2t-1)!!}{(2t)!!}$ is a decreasing function, which can be easily verified. The second inequality is due to the fact that $f(t) < \frac{1}{\sqrt{\pi t}}$. It can be derived using Stirling's formula to two orders, $n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n(1+\frac{1}{12n})$. Plug the formula into $f(t)$, then we have $f(t) = \frac{1+\frac{1}{24t}}{(1+\frac{1}{12t})^2}\frac{1}{\sqrt{\pi t}} < \frac{1}{\sqrt{\pi t}}$.

Hence, $\mathbf{E}(L2{,}1(M^T)) < \mathbf{E}(L2{,}1(M))$.

<u>**Scenario 3**: $m = 2l < n = 2k+1, l \le k$</u>

$\mathbf{E}(L2{,}1(M^T)) = m\mathbf{E}\big(Y(n)\big) = 2l\sqrt{\frac{2}{\pi}}\frac{k!k!4^k}{(2k)!}$

$\mathbf{E}(L2{,}1(M)) = n\mathbf{E}\big(Y(m)\big) = (2k+1)l\sqrt{2\pi}\frac{(2l)!}{l!l!4^l}$.

$\rightarrow \frac{\mathbf{E}(L2,1(M^T))}{\mathbf{E}(L2,1(M))} = \frac{1}{\pi(k+\frac{1}{2})}\frac{(2k)!!(2l)!!}{(2k-1)!!(2l-1)!!} \le \frac{1}{\pi\left(k+\frac{1}{2}\right)}\left(\frac{(2k)!!}{(2k-1)!!}\right)^2 = \frac{k}{k+\frac{1}{2}}\frac{(1+\frac{1}{12k})^4}{(1+\frac{1}{24k})^2} < 1.$

The second inequality can be derived by the same procedure in scenario 2.

Hence $\mathbf{E}(L2{,}1(M^T)) < \mathbf{E}(L2{,}1(M))$.

<u>**Scenario 4**: $m = 2l + 1 < n = 2k + 1, l < k$</u>

$$\mathbf{E}(L2{,}1(M^T)) = m\mathbf{E}\big(Y(n)\big) = (2l + 1)\sqrt{\frac{2}{\pi}\frac{k!k!4^k}{(2k)!}}$$

$$\mathbf{E}(L2{,}1(M)) = n\mathbf{E}\big(Y(m)\big) = (2k + 1)\sqrt{\frac{2}{\pi}\frac{l!l!4^l}{(2l)!}}.$$

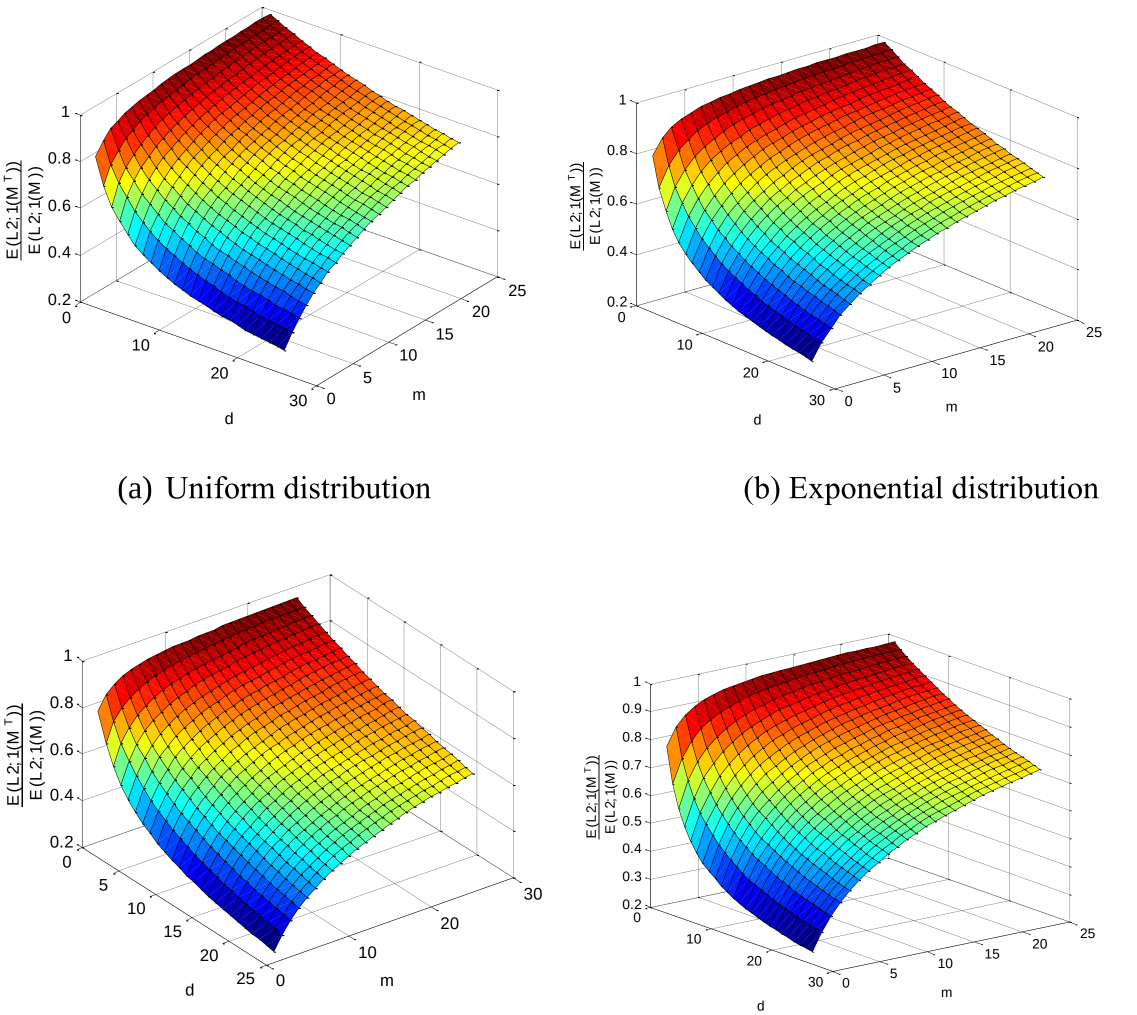
$$\rightarrow \frac{\mathbf{E}(L2{,}1(M^T))}{\mathbf{E}(L2{,}1(M))} = \frac{(2l+1)!!(2k)!!}{(2k+1)!!(2l)!!} = \frac{2k(2k-2)\ldots(2l+2)}{(2k+1)(2k-1)\ldots(2l+3)} < 1.$$

Hence $\mathbf{E}(L2{,}1(M^T)) < \mathbf{E}(L2{,}1(M))$.

In summary, $\mathbf{E}(L2{,}1(M^T)) < \mathbf{E}(L2{,}1(M))$ when $m < n$. It can be easily observed that the ratio between $\mathbf{E}(L2{,}1(M^T))$ and $\mathbf{E}(L2{,}1(M))$ increases with l and decreasing with $(m - n)$ in all four scenarios.

To validate the theoretical results, we conducted numerical experiments to evaluate the inequality of L2,1 norms between the two. In this experiment, we randomly created matrix M with m rows and n columns, where $n = m + d$. We conducted repeated experiments for each row number m and difference number d ranging from 1 to 25 under four distribution assumptions: uniform between 0 and 1, exponential with parameter 1, and normal distributions with non-zero and zero means.

Then the ratio $D(m, d) = \frac{\mathbf{E}(L2{,}1(M^T))}{\mathbf{E}(L2{,}1(M))}$ is calculated and shown is the figure below. As the figure shows, the ratio decreases with d but increases with m under all the four distribution assumptions, which is consistent with our theoretical analysis. Also we can observe that the uncertainty level of one subassembly is around 70% of that of the other when m and d are comparative (i.e., around the yellow zone in the figure below). Now we can state the findings in the lemma below.



(a) Uniform distribution (b) Exponential distribution

(c) Normal distribution with non-zero mean (d) Normal distribution with zero mean

Figure 5.5. Ratio of norms vs. row number and dimension difference

Lemma: Given a product demand standard deviation matrix M of size $m \times n$, where $M_{i,j} = \sigma_{i,j} = \bar{\sigma}|z_{i,j}|$ and $z_{i,j}$ follows i.i.d. standard normal distribution, L2,1(M^T)<

L2,1(M) when $m < n$. Please note that all the inequalities mean probabilistic inequalities. We will directly use the inequalities in the study below.

L2,1(M^T) represents the uncertainty level of the inventory of the intermediate product when only the process with m variants is completed. L2,1(M) similarly represents the uncertainty level of the inventory of the intermediate product when only the process with n variants is completed. The lemma indicates that the intermediate products with fewer variants will have less uncertainty. Up to this point we have only considered products with two points of variation. In general, industry products are much more complicated than the example products. Hence we need to generalize the lemma to make it applicable to general products.

5.2.3. Generalization

Now we can extend the discussion to a more general case, in which we have N processes and process j has v_j number of variants. Therefore M is a matrix with N dimensions. Define L2,1(M, k) = $\sum_k \sqrt{\sum_i M_{ki}^2}$, where i represents the other dimensions except for k. Then L2,1(M, k) can represent the uncertainty level of the inventory of the intermediate product when only process k is finished. Please note that k can be a dimension with any length, so is its complementary dimension i.

The lemma hence can be recast as: given a standard deviation matrix M of N dimensions, whose elements follow an i.i.d. normal distribution, L2,1(M, i)< L2,1(M, j) when $v_i < v_j$, where v_i is the size of dimension i (the number of variants of process i).

Theorem 1: L2,1(M, k) < L2,1(M, l) if $v_k < v_l$, given a standard deviation matrix M, which is an N dimensional matrix with each element following i.i.d. normal distribution.

Proof: L2,1(M, k) = $\sum_k \sqrt{\sum_i M_{ki}^2}$ where $\sum_i M_{ki}^2 = \sum_l \sum_j M_{klj}^2$, where j represents the other elements except for k and l. Similarly, L2,1(M, l) = $\sum_l \sqrt{\sum_i M_{li}^2}$, where $\sum_i M_{li}^2 = \sum_l \sum_j M_{klj}^2$. Denote $\sum_j M_{klj}^2$ as m_{kl}. Then L2,1(M, k)= $\sum_k \sqrt{\sum_l m_{kl}^2}$ < L2,1(M, l)= $\sum_l \sqrt{\sum_k m_{kl}^2}$, following directly from the lemma.

A graphical explanation is shown in Figures 5.6 and 5.7 where L2,1 norm is applied along dimension 1. Blocks in the figures represent product variation and the vectors represent the dimensions. Given a dimension along which to apply L2,1 norm, the L2,1 norm will sum the L2 norm of the matrix perpendicular to the assigned dimension (in orange). In a 2D matrix, the perpendicular matrix is a vector while in a 3D matrix the perpendicular matrix is a 2D matrix as shown in Figures 5.6 and 5.7.

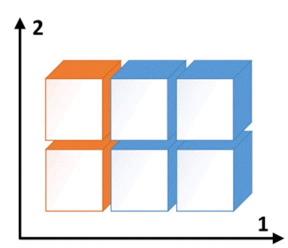


Figure 5.6. Graphical illustration of 2D L2,1 norm

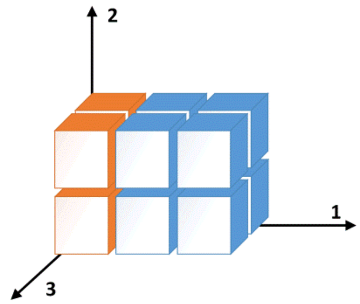


Figure 5.7. Graphical illustration of 3D L2,1 norm

The theorem indicates that the uncertainty level of a subassembly is positively correlated to the number of variants of that subassembly. Since safety inventory is linearly proportional to uncertainty level, the number of variants of a subassembly is a good reflection of its safety inventory. If we can further find a linear indicator of the uncertainty level, that indicator will well serve as a measure of safety inventory.

5.2.4 Measure of Uncertainty due to Variety

The uncertainty level of a subassembly with processes k completed is denoted as $L2{,}1(M,k) = \sum_k \sqrt{\sum_i M_{ki}^2}$. Denote $Y(t) = \sqrt{\sum_i z_i^2}$, hence $\mathbf{E}(L2{,}1(M,k)) = \bar{\sigma} v_k \mathbf{E}(Y(v_i))$, where $v_k v_i = V$, and V is the total number of variants of the final product. Applying asymptotic approximation $\mathbf{lim}_{t\to\infty} \frac{\Gamma(t+\frac{1}{2})}{\Gamma(t)\sqrt{t}} = 1$ to $\mathbf{E}\big(Y(t)\big) = \sqrt{2}\frac{\Gamma(\frac{t+1}{2})}{\Gamma(\frac{t}{2})}$, we have $\mathbf{E}(L2{,}1(M,k)) \sim \bar{\sigma} v_k \sqrt{2v_i} = \bar{\sigma}\sqrt{2V}\sqrt{v_k}$. Hence $\sqrt{v_k}$ is a linear measure of the safety inventory level of subassembly with processes k finshed.

Theorem 2: $\mathbf{E}(L2{,}1(M,k)) \sim \bar{\sigma}\sqrt{2V}\sqrt{v_k}$, given a standard deviation matrix M, which is an N dimensional matrix with each element following i.i.d. normal distribution and $V = \prod_{i=1}^{N} v_i$.

In summary, this section proves that the number of variants of a subassembly is positively correlated to its safety inventory level. In addition, the square root of its number of variants can serve as a linear measure of the safety inventory level. Now we can apply the results to two problems we discussed in the beginning of this section, i.e., process sequencing and assembly decomposition problems.

5.3 Process Sequencing Optimization

Consider a product requiring N processes, where process k has v_k variants. There are in total $\prod_{i=1}^{N} v_i$ final products. Each product can be denoted in an array $(a_1, a_2, \dots, a_N)$, where a_i represents the variant of process i, i.e., $1 \le a_i \le v_i$. Define a function $f: \mathbb{N}^N \to \mathbb{N}$ that can transform the array into a unique number $j \in [1, \prod_{i=1}^{N} v_i]$. The distribution of product variant $(a_1, a_2, \dots, a_N)$, is denoted as normal distribution (μ_j, σ_j) where $j = f(a_1, a_2, \dots, a_N)$. Define M as the standard deviation matrix, $M \in \mathbb{R}^N$ and $M(a_1, a_2, \dots, a_N) = \sigma_j$, where $j = f(a_1, a_2, \dots, a_N)$.

Starting from the raw materials or base products, a buffer or inventory will be held for the processed products right after each process. Excluding the final product, there will be $N - 1$ intermediate products. A sequence can be denoted as a string of processes $(P_1, P_2, \dots, P_k \dots, P_N)$, which is a permutation of the processes 1 to N. For example, $(P_1 = 1, P_2 = 2, \dots, P_k = k, \dots, P_N = N)$ represents an ordered sequence to execute the processes from process 1 to N. The safety inventory for the intermediate product at step i can thus be represented with $zL_{2,1}(M, s_i)$, where s_i is the array of the subscript until step i. Our objective is to identify a strategy to design the process sequence so as to minimize the total inventory of all the intermediate products. Since inventory levels of subassemblies are determined by their safety inventories, we can simplify our objective as minimizing the total safety inventory of the all the intermediate products.

For the purpose of illustration, we assume that there are no precedence constraints between the processes. We will prove that the optimal strategy is to delay the processes with the largest number of variants as late as possible so that the number of variants is non-increasing from stage 1 to stage N, i.e., $(v_{P_1}, v_{P_2}, \ldots, v_{P_k}, \ldots, v_{P_N})$ is a non-increasing sequence.

We will derive a corollary to prove our hypothesis.

Corollary: L2,1(M, (s, k))< L2,1(M, (s, l)) if $v_k < v_l$.

The corollary follows directly from **Theorem 1** by expanding the dimensions.

Given a sequence $(P_1^*, P_2^*, \ldots, P_{j-1}^*, P_j^*)$ with a non-decreasing sequence in terms of the number of variants, we will prove that swapping any two positions with different variants will increase the total safety inventory level. There are two scenarios when swapping two positions. If we swap two positions next to each other, P_j^* and P_{j+1}^*, i.e., the new sequence is $(P_1^*, P_2^*, \ldots,\ P_{j-1}^*,\ P_{j+1}^*,\ P_j^*, \ldots, P_N^*)$. Then the inventory levels except stage *j* remain the same. In the original sequence, the safety inventory at stage j is $zL_{2,1}(M, (P_1^*, P_2^*, \ldots, P_{j-1}^*, P_j^*))$, while the second has safety inventory at stage j of $zL_{2,1}(M, (P_1^*, P_2^*, \ldots, P_{j-1}^*, P_{j+1}^*))$. Define s= $(P_1^*, P_2^*, \ldots, P_{j-1}^*)$, $k = P_j^*$, $l = P_{j+1}^*$, since

$v_{P_j^*} < v_{P_{j+1}^*}$, by the corollary, $L2,1(M,\ (s,k)) < L2,1(M,\ (s,l))$. Swapping the two positions increases the total inventory level. In the second scenario, we swap two positions P_i^* and P_j^* at least one position far away from each other, where $j > i + 1$ and$v_{P_i^*} < v_{P_j^*}$. Similarly, only the inventory levels between stage i and $j - 1$ change. By applying the corollary $j - i - 1$ times, we can show that the inventory at each stage between the two swapping stages increases. In summary, swapping two positons so that the one with smaller number of variants is swapped to a later stage will increase the total inventory level.

By induction, we can show that the optimal strategy is to create a sequence with non-decreasing number of variants to minimize the total safety inventory level. We call this strategy prioritized differentiation, i.e., prioritize the sequence by its number of variants.

Based on the theorem on the two-process case, we develop a guideline to optimize process sequences, which is to delay the processes with the large number of variants as late as possible. This is a generalized study with the focus on safety inventory level. We can regard delayed differentiation as a special case of our study. Please note that the process referred to in this dissertation is a general concept. It might be a collection of processes in facilities. In that case, the inventory is the one between facilities. The results provide practitioners and researchers a general guideline. In addition, the impact of the sequencing will be more exaggerating as the processes are becoming more and more complex as illustrated in the discussion above. Readers do not have to limit our work to make-to-stock scenarios, although safety inventory is a concept mainly for make-to-stock strategies. However, in make-to-order scenarios, we may find other areas that still have risk pooling opportunities such as the uncertainties in orders. Combined with powerful database and other techniques, the results may find more applications in real industrial settings.

5.4 Assembly Decomposition Optimization

This section will focus on the assembly decomposition so as to minimize the total inventory. Since we have a measure to represent the safety inventory level at each stage, it will be equivalent to minimizing the total number of variants at each stage. The idea is similar to Process Sequence Graph (PSG) in [61], in which they minimized the total number of nodes in PSG. They claimed that "our work at HP is currently attempting to quantify how the reduction in nodes might actually affect cost". Another limitation of this work is that the algorithm proposed can only handle serial sequence planning while nowadays parallel planning is common. The DP approach to the integrated optimization of assembly plans and supply chain configurations [73] can be adjusted for the optimization in this chapter. A formal problem statement and algorithm description will be given below. For simplicity, this part will only consider assembly decomposition incurred variety.

Assume the manufacture of assembly product $\mathbf{P} = (\mathbf{C}, \mathbf{L})$, with N components and K assembly processes, where $\mathbf{C}$ is the set of product components ($\mathbf{C} = \{c_1, c_2, \ldots c_N\}$) and $\mathbf{L}$ is the set of assembly processes ($\mathbf{L} = \{l_1, l_2, \ldots l_K\}$).

Then, the AOG $\mathbf{H} = (\mathbf{V}, \mathbf{A})$ for product $\mathbf{P}$ is as follows:

$\mathbf{V} = \{v_i \mid i \in \mathbf{I_v}\}$, where v_i represents a subassembly. The number of vertices in $\mathbf{H}$ is $|\mathbf{I_v}|$. Define $M = |\mathbf{I_v}| - N$. The number of CVs in $\mathbf{H}$ is M; then the total number of vertices in

$\mathbf{H}$ is $M+N$. We label vertices from lower to higher levels in the ascending order using function $leb(v)$ so that $i > j$ if the level of v_i is higher than that of v_j. Hence, the SVs are labeled from 1 to N and the CVs from $N+1$ to $M+N$.

$\mathbf{A} = \{\mathbf{a}_i \mid i \in \mathbf{I_a}\}$, where $\mathbf{a}_i = (v_p^i, v_l^i, v_r^i)$. Define function $fapi(i)$ as the total number of variants of assembling v_l^i and v_r^i into v_p^i.

Our objective is to choose the assembly plan out of the feasible plans to minimize the total measure.

5.4.1 Optimization Algorithm

The objective is $\min\limits_{\mathbf{a},q}(tn(v_{M+N} \mid \mathbf{a}))$

Subject to

$\mathbf{a} \in \mathbf{A}_{M+N}$ and $q \in \mathbf{Q}_l$, in which $l = fapi(\mathbf{a})$

A summary of the DP algorithm is provided below.

State: (i,k), where i ($N+1 \le i \le N+M$) is the index of CV v_i; k ($k \le d_i$) is the AND relation index of vertex v_i;

Value Function: $V(i,k)$ is defined as the total measure of CV v_i given the assembly method k.

Functional Equation:

$V(i,k) = \min\limits_{k_1;k_2}(V(i_1,k_1) + V(i_2,k_2) + tn(i,k))$, where

$i_1 = leb(\mathbf{a}_i(2))$, $k_1 \le d_{i_1}$, $l_1 = fapi(\mathbf{b}_{i_1}(k_1))$;

$i_2 = leb(\mathbf{a}_i(3))$, $k_2 \le d_{i_2}$, $l_2 = fapi(\mathbf{b}_{i_2}(k_2))$;

$k \le d_i$, $l = fapi(\mathbf{b}_i(k))$.

Boundary Condition: $V(i,k) = tn(i)$, $k = 1$, $s \in \mathbf{S}_i$ when $i \le N$.

The optimal measure is $V = \min_k V(M+N,k)$.

5.4.2 Computational Complexity

This section discusses the theoretical and practical computational complexities of the solution algorithm. A similar analysis appears in Chapter 3. An upper bound for the computational complexity for this algorithm is $O(MD^2)$, where M is the number of CVs in the HAG, D represents the maximum value for the numbers of feasible options to build a subassembly,. In common products, $M = O(N^2)$ and $D = O(N)$, where N is the number of the components constituting the product. Hence the computational complexity of the DP algorithm for practical products is $O(N^4)$.

5.5 Case Study

We use Boujault's pen assembly in Chapter 3 as an example to illustrate the product supply chain optimization problem. The AND/OR graph of the example pen is shown in Chapter 3. All the six feasible assembly plans are listed in Figure 5.8 below.

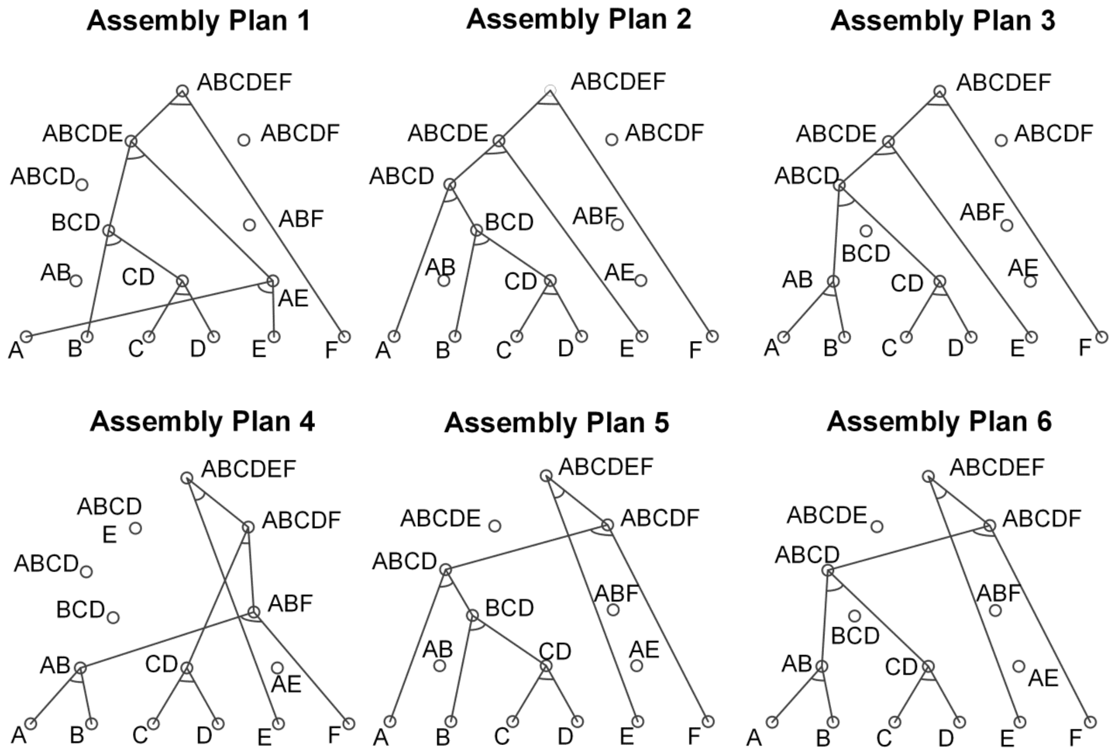


Figure 5.8. Six assembly plans of the pen

To satisfy the needs from customers, the company provides a wide selection of features of the pen. There are 3 choices of head (B: 0.2mm, 0.3mm, and 0.5mm), 6 colors of body (A: black, white, red, yellow, green, and blue), 3 colors of ink (D: red, black, and blue), and 3 types of caps (F: wooden, plastics, and metal). The total measures of each plan are respectively 10.0, 13.5, 14.4, 18.2, 17.3, and 18.2. The optimal assembly plan in this case is plan 1. This is because that it delays the processes A and F until a later stage comparing with other plans. Due to the assembly constraints, the last subassembly must be from ABCDF or ABCDE. According to our prioritization strategy, it is wise to delay F, hence ABCDE is chosen as the last subassembly, i.e., plan 4, 5, and 6 are ruled out. The number of variants of ABCDE is 54, which could be assembled through ABCD (54) and E (1) or AE (6) and BCD (9). Obviously decomposing ABCDE into two balanced subassemblies will help reduce the uncertainty level. In addition to the height-balance introduced in Chapter 4, the degree of balance of sub-assemblies of a product family reflects the difference between the varieties of their branches. A balanced sub-assembly tends to have two branches with similar numbers of varieties. Hence assembly plan 1 is the optimal one. The example shows us how the algorithm could be utilized to optimize the assembly plans so that inventory is minimized. It also provides practitioners with strategies and insights into to assembly supply chain designs.

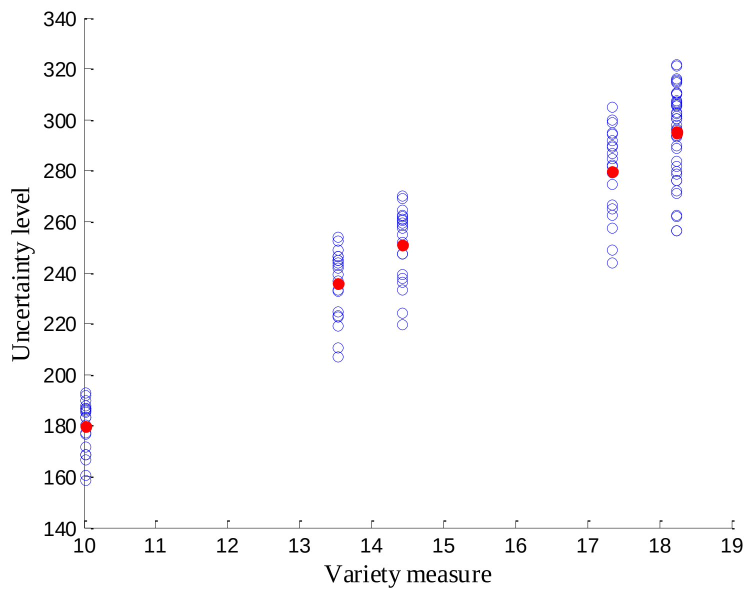


Figure 5.9. Uncertainty level vs. variety measure with a sample size of 20

In the figure above, the uncertainty level of all the samples from a Monte Carlos simulation are plotted in blue circles and the average of all the 20 experiments are plotted in red dots. The average inventory level for all the six assembly supply chains is around 240 while the lowest is less than 180. A careful design of assembly supply chains can help achieve a 25% reduction compared to an average case if not the worst one.

5.6 Summary

This chapter studied the impact of assembly sequence on safety inventory when customer demand is unknown and the product variety is high. Assuming the variation matrix follows normal distributions, we showed the structural difference lying in the assembly supply chain designs. We proposed a measure to reflect the efficiency of the assembly supply chain. Unlike other measures, which have limited connection with real indices, such as average production cost and lead time, the relation between the proposed measure and safety inventory was presented explicitly. The theorem generalized the delayed differentiation strategy to all products regardless of the commonality shared throughout the product family, which we call prioritized differentiation. A DP based algorithm was provided to search for the optimal assembly plans with the lowest inventory. A comparison between the proposed measure and the average inventory was conducted using Monte Carlo simulation. The numerical results confirmed the theorem, i.e., the measure and total safety inventory are positively correlated. The case studies illustrated the strategies of periodization and balancing.

Through this study, we provided a fundamental investigation on product family assembly supply chain designs under uncertainty and high variety. The measure of the impact of the product variety on uncertainty related costs gives practitioners a concrete idea in the design practices. They may use the measure to approximate the uncertainty related costs such as tooling costs of machines, inventory costs in warehouses, and ergonomic concerns in assembly lines. Based on such quantitative approximation, they may further make decisions on how to arrange the sequences of manufacturing processes or supply chains. Our study provides a fundamental understanding of the impact of assembly planning on supply chains. The benefits of the generalized delayed differentiation, or prioritized differentiation, in inventory reduction are also shown in this chapter. A potential of 30~ 40% reduction of uncertainty in an assembly supply chain is shown through the numerical studies. In combination with other considerations such as quality control and lead time management, more useful guidelines can be generated.

CHAPTER 6

Conclusions and Future Work

This chapter summarizes the contributions of this dissertation and discusses directions of future work.

6.1 Conclusions and Contributions

This dissertation presents the original research work on integrated assembly plan and supply chain designs ranging from models, algorithms, to strategies under two scenarios: single product and product family. The contributions are summarized below.

- A novel model has been established to integrate assembly plans with supply chain configurations by grouping the assembly nodes into the Hyper AND/OR graph (HAG). HAG is shown to be an efficient model to represent the supply chain information and assembly constraints. HAG may find various applications where nonlinear constraints are interwoven with network information.
- A bottom-up DP algorithm with polynomial time complexity has been developed to search for the global optimal assembly plan and supply chain configuration with the lowest total cost. The revised DP algorithm can solve the problem with a lead time constraint by introducing an extra state variable - lead time level. The computational complexity of the algorithm for the time constrained problem is

pseudo-polynomial. Comparing with enumerative or heuristic algorithms, our method exhibits the advantages of structural searches.

- The case studies for single products illustrated the applications of our method and the insights designers may learn out of the results. Through better understanding of the interdependence between suppliers and assembly plans, concurrent decisions can be made to shorten the product development lead time and to lower the total supply chain cost. Numerical results showed that optimal designs yield a 20% reduction of cost and 10 % reduction in lead time in the given case study. The results also explained the merits of industrial clustering and branch balancing strategies. The method to coordinate the assembly planning and supply chain design enables the responsive adaptation of assembly and supply chain systems when companies face supply and/or technical changes.

- The impact of variety on inventory was quantified and a useful measure was proposed for the design of assembly supply chains considering demand uncertainty. This measure allows assembly planning and supply chain practitioners to build a quantitative sense on variety incurred inventory. Such a sense will benefit practitioners from various perspectives ranging from inventory control, work shop scheduling, to ergonomics.

- A generalized "delayed differentiation" (DD) strategy suggested, which is called prioritized differentiation (PD) in this dissertation. This PD strategy suggests delaying the processes or subassemblies with more variants to later stages when multiple options of variation exist. The significance of the PD strategy lies in the fact that it assumes no commonality through the product family as DD does, which makes its scope of applications much wider in nowadays' manufacturing industry.

- A synthesis of models, algorithms, and strategies under single product and product family scenarios provides researchers and industrial participants a systematic understanding of the interaction between assembly plans and supply chains configurations.

6.2 Future Work

The future directions are discussed in the previous chapters. A quick summary of the potential future work is listed below.

- Apply hyper AND/OR graph to different problems. As discussed in Chapter 2, HAG provides a good platform to represent nonlinear constraints and resource availability. It has good potential to be applied to different environments, such as logistics optimization and in-facility routing optimization.
- Explore the possibility of modeling the integrated design problem as a shortest path problem (SPP) through utilizing state diagrams as discussed in Chapter 2. By modeling the problem as an SPP, all the existing SPP algorithms may be applied to improve the efficiency.
- Improve the efficiency of the algorithm developed for the time constraint optimization problem in Chapter 4 through applying a tighter bound. As explained in Chapter 4, the developed algorithm relaxed the time bounds to the upper bound, which left much space for further optimization.

- Consider the correlations between the products in a family, which is not uncommon in industries. Taking the customized pen as an example, customers tend to match the color of the ink with that of the cap so that they can make an easy recognition. Under such circumstance, the variety of certain products may not have as much impact on their inventory as the others. How the strategy should be adjusted to certain conditions requires further investigation.

Bibliography

[1] A. Gunasekaran, "An integrated product development-quality management system for manufacturing," *The TQM Magazine*, vol. 10, no. 2, pp. 115-123, 1998.

[2] Y. K. Ro, J. K. Liker and S. K. Fixson, "Modularity as a strategy for supply chain coordination: The case of U.S. auto," *IEEE Transactions on Engineering Management*, vol. 54, 2007.

[3] Ganapathy, B.K. and C.-H. Goh, "A hierarchical system of performance measures for concurrent engineering". *Concurrent Engineering*, 1997. 5(2): p. 137-143.

[4] A. Bourjault, "Contribution a une approche methodologique del'sassemblage automatise: elaboration automatique des sequences operations," 1984.

[5] L. Homem De Mello and A. Sanderson, "AND/OR Graph representation of assembly plans," *IEEE Transactions on Robotics and Automation*, vol. 6, 1990.

[6] J. Wolter, "A combinatorial analysis of enumerative data structures for assembly planning," *Journal of Design and Manufacturing*, vol. 2, pp. 93-104, 1992.

[7] L. Homem De Mello and A. Sanderson, "Representation of mechanical assembly sequences," *IEEE Transactions on Robotics and Automation*, vol. 7, 1991.

[8] D. Whitney, Mechanical Assemblies: Their Design, Manufacture, and Role in Product Development, New York: Oxford University Press, 2004.

[9] A. Sanderson and L. S. Homem De Mello, "A correct and complete algorithm for the generation of mechanical assembly sequences," *IEEE Transactions on Robotics and Automation*, vol. 7, pp. 228-240, 1991.

[10] N. Ong and Y. Wong, "Automation subassembly detection from a product model for disassembly sequence generation," *International Jounral of Advanced Manufacturing Technology*, vol. 15, pp. 425-431, 1999.

[11] P. Knosla and R. Mattikali, "Determing the assembly sequence from a 3-D model," *Journal of Mechanical Working Technology*, vol. 20, pp. 153-162, 1989.

[12] J. Gao, D. Xiang and G. Duan, "Subassembly indentification based on grey clustering," *International Jounal of Production Research*, vol. 46, 2008.

[13] S. Lee, "Subassembly Indentification and Evaluation for Assembly Planning," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, 1994.

[14] S. Li, H. Wang, S. Hu, Y.-T. Lin and J. Abell, "Automatic generation of assembly system configuraton with equipment selection for automotive battery manufacturing," *Journal of Manufacturing System*, vol. 30, pp. 188-195, 2011.

[15] J. Ko, E. Nazarian, H. Wang and J. Abell, "An assembly decomosition model for subassembly plannng considering imperfect inspection to reduce assembly defect rates," *Journal of Manufacturing Systems*, vol. 32, no. 3, pp. 412-416, 2013.

[16] N. Fujimoto, N. Fuji and Y. Nagata, "A new sequence evolution apporach to assembly planning," *ASME Journal of Manifacturing Science and Engineering*, vol. 122, pp. 198-205, 2000.

[17] Z. Liang and M. Xu, "Concurrent optimization of product module selection and assembly line configuration: A multi-objective approach," *ASME Jounral of Manufacturing Science and Engineering*, vol. 127, pp. 875-884, 2005.

[18] R. Oliver and M. Webbler, Supply chain management: logistics catches up with strategy, London: Outlook, Booz, Allen and Hamilton Inc., 1992.

[19] J. Shapiro, Modeling the Supply Chain, Pacific Grove, CA: Wadsworth Group, 2001.

[20] S. Graves and S. Willems, "Optimizing the supply chain configuration for new products," *Management Science*, vol. 51, pp. 1165-1180, 2005.

[21] U. Thonemann and J. Bradley, "The effect of product variety on supply chain performance," *European Jounral of Operational Research*, vol. 143, pp. 548-556, 2002.

[22] N. Viswanadham and R. Gaonkar, "Partner selection and synchronized planning in dynamic manufacturing networks," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 117-130, 2003.

[23] S. Torabi and E. Hassini, "An interactive possibilistic programming approach for multiple," *Journal of Fuzzy Sets and Systems*, no. 159, pp. 193-214, 2008.

[24] A. Gunasekaran, K.-H. Lai and T. E. Cheng, "Responsive supply chain:Acompetitive strategy in a networked economy," *The international journal of management science*, vol. 36, pp. 549-564, 2008.

[25] O. Willianson, "Outsourcing: Trasaction cost economics and supply chain management," *Journal of Supply Chain Management*, vol. 44, no. 2, pp. 5-16, 2008.

[26] S. E. Fawcett, G. M. Magnan and M. W. McCarter, "Benefits, barriers, and bridges to effective supply chain management," *Supply Chain Management: An International Journal*, vol. 13, no. 1, pp. 35-48, 2008.

[27] I. Manuj and J. T. Mentzer, "Global supply chain risk management strategies," *International Journal of Physical Distribution & Logistics Management*, vol. 38, no. 3, pp. 192-223, 2008.

[28] S. McGovern and S.M. Gupta, The Disassembly Line, Balancing and Modeling, The McGraw-Hills Companies Inc, 2011.

[29] M. Rungtusanatham and C. Forza, "Coordinating product design, process design, and suply chain design decision Part A: Topic motivation, performance implications, and article review process," *Journal of Operational Management*, pp. 257-265, 2005.

[30] C. Forza, F. Salvador and M. Rungtusanatham, "Coordinating product design, process design, and supply chain design decison Part B: Coordinationg approaches, tradeoffs, and future research directions," *Jounal of Operational Management*, pp. 257-265, 2005.

[31] G. Huang, X. Zhang and L. Liang, "Towards integrated optimal configuration of platform products, manufacturing prcesses, and supply chain," *Jounal of Operations Management*, vol. 23, pp. 389-403, 2005.

[32] C. Fine, B. Golany and H. Naseraldin, "Modeling tradeoffs in three-dimensional concurrent engineering: a goal programming approach," *Journal of Operaitons Management*, vol. 23, pp. 345-369, 2005.

[33] S. Fixson, "Product architecture assessment: a tool to link product , process, and supply chain design decisions," *Journal of Operations Management*, vol. 23, pp. 345-369, 2005.

[34] Shao, X., et al., "Integration of process planning and scheduling—a modified genetic algorithm-based approach". *Computers & Operations Research*, 2009. 36(6): p. 2082-2096.

[35] Che, Z.H., "A genetic algorithm-based model for solving multi-period supplier selection problem with assembly sequence". *International Journal of Production Research*, 2010. 48(15): p. 4355-4377.

[36] Che, Z.H. and C.J. Chiang, "A modified Pareto genetic algorithm for multi-objective build-to-order supply chain planning with product assembly". *Advances in Engineering Software*, 2010. 41(7-8): p. 1011-1022.

[37] S. Hu and J. Ko, "Assembly systm design and oprations for product variety," *CIRP Annals - Manufacturing Technology*, vol. 60, pp. 715-733, 2011.

[38] Christofides, N., Graph theory: An algorithmic approach. Vol. 8. 1975: Academic press New York.

[39] Garey, M. and D. Johnson, Computers and Intractability: A Guide to the Theory of Incompleteness, 1979, HW Freeman and Company, San Francisco.

[40] Beasley, J. and N. Christofides, "An algorithm for the resource constrained shortest path problem". Networks, 1989. 19(4): p. 379-394.

[41] Joksch, H.C., "The shortest route problem with constraints". *Journal of Mathematical analysis and applications*, 1966. 14(2): p. 191-197.

[42] Hindelang, T.J. and J.F. Muth, "A dynamic programming algorithm for decision CPM networks". *Operations Research*, 1979. 27(2): p. 225-241.

[43] Robinson, D.R., "A dynamic programming solution to cost-time tradeoff for CPM". *Management Science*, 1975. 22(2): p. 158-166.

[44] Möhring, R.H. and F.J. Radermacher, "Substitution decomposition for discrete structures and connections with combinatorial optimization". *North-Holland mathematics studies*, 1984. 95: p. 257-355.

[45] Feillet, D., et al., "An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems". Networks, 2004. 44(3): p. 216-229.

[46] Davis, E.W. and J.H. Patterson, "A comparison of heuristic and optimum solutions in resource-constrained project scheduling". Management science, 1975. 21(8): p. 944-955.

[47] Handler, G.Y. and I. Zang, "A dual algorithm for the constrained shortest path problem". Networks, 1980. 10(4): p. 293-309.

[48] Bouleimen, K. and H. Lecocq, "A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version". *European Journal of Operational Research*, 2003. 149(2): p. 268-281.

[49] Hartmann, S., "A competitive genetic algorithm for resource ‐ constrained project scheduling". Naval Research Logistics (NRL), 1998. 45(7): p. 733-750.

[50] Hartmann, S. and R. Kolisch, "Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem". *European Journal of Operational Research*, 2000. 127(2): p. 394-407.

[51] Avella, P., M. Boccia, and A. Sforza, "A penalty function heuristic for the resource constrained shortest path problem". *European Journal of Operational Research*, 2002. 142(2): p. 221-230.

[52] Merkle, D., M. Middendorf, and H. Schmeck, "Ant colony optimization for resource-constrained project scheduling". *Evolutionary Computation, IEEE Transactions* on, 2002. 6(4): p. 333-346.

[53] Fonte, William Giacomo. "A de-proliferation methodology for the automotive industry." PhD diss., Massachusetts Institute of Technology, 1994.

[54] Monk, E., and Wagner, B., 2012, "Concepts in enterprise resource planning", Cengage Learning.

[55] Levi, D., Kaminsky, P., and Simchi-Levi, E., 2003, "Chapter 3: Inventory Management and Risk Pooling," Designing & Managing the Supply Chain-Second Edition"(p-66).

[56] Lee, H. L., 2002, "Aligning supply chain strategies with product uncertainties," *California management review*, 44(3), pp. 105-119.

[57] Deshmukh, A. V., Talavage, J. J., and Barash, M. M., 1998, "Complexity in manufacturing systems, Part 1: Analysis of static complexity," IIE transactions, 30(7), pp. 645-655.

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

[59] Kuzgunkaya, O., and ElMaraghy, H. A., 2006, "Assessing the structural complexity of manufacturing systems configurations," *International Journal of Flexible Manufacturing Systems*, 18(2), pp. 145-171.

[60] Wang, H., and Hu, S. J., 2010, "Manufacturing complexity in assembly systems with hybrid configurations and its impact on throughput," CIRP Annals-Manufacturing Technology, 59(1), pp. 53-56.

[61] Martin, M. V., and Ishii, K., "Design for variety: development of complexity indices and design charts," *Proc. Proceedings* of, pp. 14-17.

[62] Modrak, V., and Marton, D., 2013, "Development of metrics and a complexity scale for the topology of assembly supply chains," *Entropy*, 15(10), pp. 4285-4299.

[63] Van Hoek, R. I., 2001, "The rediscovery of postponement a literature review and directions for research," *Journal of operations management*, 19(2), pp. 161-184.

[64] Yang, B., and Burns, N., 2003, "Implications of postponement for the supply chain," *International Journal of Production Research*, 41(9), pp. 2075-2090.

[65] Yang, B., Burns, N. D., and Backhouse, C. J., 2004, "Postponement: a review and an integrated framework," *International Journal of Operations & Production Management*, 24(5), pp. 468-487.

[66] Yang, B., and Yang, Y., 2010, "Postponement in supply chain risk management: a complexity perspective," *International Journal of Production Research*, 48(7), pp. 1901-1912.

[67] Lee, H. L., and Tang, C. S., 1997, "Modelling the costs and benefits of delayed product differentiation," *Management science*, 43(1), pp. 40-53.

[68] Swaminathan, J. M., and Tayur, S. R., 1998, "Managing broader product lines through delayed differentiation using vanilla boxes," *Management Science*, 44(12-part-2), pp. S161-S172.

[69] Gupta, S., and Krishnan, V., 1998, "Product family-based assembly sequence design methodology," *IIE transactions*, 30(10), pp. 933-945.

[70] Forza, C., Salvador, F., and Trentin, A., 2008, "Form postponement effects on operational performance: a typological theory," *International Journal of Operations & Production Management*, 28(11), pp. 1067-1094.

[71] Blecker, T., and Abdelkafi, N., "Modularity and delayed product differentiation in assemble-to-order systems: analysis and extensions from a complexity perspective," *Proc. Proceedings Int. Mass Customization Meeting 2005*, Gito Verlag Berlin, pp. 29-46.

[72] Ding, C., Zhou, D., He, X., and Zha, H., "R 1-PCA: rotational invariant L 1-norm principal component analysis for robust subspace factorization," *Proc. Proceedings of the 23rd international conference on Machine learning*, ACM, pp. 281-288.

[73] Kuang, H., S.J. Hu, and J. Ko, "Concurrent design of assembly plans and suply chain configurations using AND/OR graphs and dynamic programming", *ASME Jounral of Manufacturing Science and Engineering* (in press), 2015.