GAME THEORETIC OPTIMIZATION FOR HIGH VARIETY ASSEMBLY SYSTEM DESIGN

A Dissertation
Presented to
The Academic Faculty

by

Yitao Liu

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Mechanical Engineering

Georgia Institute of Technology
December 2016

GAME THEORETIC OPTIMIZATION FOR HIGH VARIETY ASSEMBLY SYSTEM DESIGN

Approved by:

Dr. Roger J. Jiao, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Seung-Kyum Choi
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Yan Wang
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Katherine Fu
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Manpreet S. Hora
Scheller College of Business
Georgia Institute of Technology

Date Approved: August 25, 2016

ACKNOWLEDGMENTS

It would have been not possible for me to finish my doctoral dissertation without the help and support of the people around me, the excellent research environment provided by, and generous funding supported by Georgia Institute of Technology.

First and foremost, I would like to take this opportunity to express my sincerest gratitude and appreciation to my advisor Dr. Roger J. Jiao at Georgia Institute of Technology for his invaluable guidance, supervision, and advice for my Ph.D. study. I am greatly appreciated for Dr. Roger J. Jiao who helps shape my research ability and carries me on through difficult times with his insights and suggestions.

Special thanks go to my thesis reading committee, including Dr. Seung-Kyum Choi (Mechanical Engineering), Dr. Yan Wang (Mechanical Engineering), Dr. Katherine Fu (Mechanical Engineering), and Dr. Manpreet S. Hora (Scheller College of Business) for their precious time and suggestions to improve my dissertation in various ways.

I would like the thank Renewable Bioproduct Institute (RBI/IPST) of Georgia Tech on their fellowship support to my PhD research. Specifically, I would like to thank Dr. Norman F. Marsolan, Dr. Jeffery S. Hsieh, Dr. Yulin Deng, Lavon Harper and Bob Davies for their help in my research and career advising.

I also would like to thank all the students in Room 264B MaRC building, staff at School of Mechanical Engineering at Gatech, and colleagues in Tyco Electronics, who always are ready to help me in times of need. Specifically, I would like to thank Dr. Feng Zhou, Dr. Yangjian Ji, Ruoyu Song, Tianyi Li, Chad Hume, Recep M. Gorguluarslan, Glenda Johnson, Trudy Allen, Olivia Kulisz, Dr. Roberto Lu, Michael F. Laub, Sara

Bolha, Dr. Parag Patre, Peter J. Cherok, Dr. Huadong Wu, Swapnilsinh Solanki, Sezai Sablak, Charles R. Malstrom, and Yixiang Yan.

Last but not least, great thanks go to my family for always being there when I need them most, and for supporting me through all these years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

Chapter 4

F_U	Leader optimization function in game theoretic optimization
F_L	Follower optimization function in game theoretic optimization
$\mathbf{X} \in R^s$	A *s*-dimensional design variable
$\mathbf{Y} \in R^t$	A *t*-dimensional design variable
$\mathbf{Y}^*$	An optimal solution of leader's perception of follower's decision
G_U	A *p*-dimensional vector valued function
G_L	A *q*-dimensional vector valued function
q_1	Leader's production quantity
q_2	Follower's production quantity
a, b, c_1, c_2	Coefficients of linear price model
π	Economic profit function

Chapter 5

$\mathbf{X}_k$	*k*-th product variant
$\mathbf{Y}_k$	Process plan of *k*-th product variant
m_k	Number of work stations involved in assembly of product variant *k*
$f(s)$	Fitness value of solution *s*
$\overline{f}$	Average fitness

Chapter 6

$\{R_i\}$	Raw material
$\{C_i\}$	Purchased part
$\{I_i\}$	Intermediate part
$\{SA_i\}$	Subassembly
$\{SR_i\}$	Structure relationship

$\{IC_i\}$	Include condition
$\{A_i\}$	Assembly operation
$\{M_i\}$	Machining operation
W	Workstation
T	Cycle time
S	Setup number
$X=\{V_i^*\}$	Process variety instances,
a_j	Process type instances
M_k	Machine instances
T_s	Process tooling instances
F_t	Process fixture instances
$s\%$	Support level
$c\%$	Confidence level
$\Psi(I^G, O^G)$	Generic relationship mapping

Chapter 7

$PC_{ls}, PC_{sf}, PC_{ef}$	Process indices
F	Process commonality index (Tsubone et al., 1994)
$CI^{(P)}$	Process commonality index (Jiao and Tseng, 2000)
PCI	Product line commonality index
CMC	Comprehensive metric commonality
x_{ia}	a-th process plan of part i
w	Number of feasible assembly process plans
n_l	Number of instances for the l-th resource type
G	Number of resource clusters
$g_m=(\tilde{g}_{1m},\cdots\tilde{g}_{lm},\cdots\tilde{g}_{Lm})$	m-th resource cluster
$\mathbf{X}_r$	r-th assembly process plan
$\mathbf{Y}_{rs}=\left(y_{rs1}, y_{rs2}, \cdots y_{rsi}, \cdots y_{rsn_d}\right)$	r-th resource allocation plan

$v \in N^+$	Number of feasible resource allocation plans
C	Process flexibility index
F	Process sequencing flexibility index
A	Process feasibility index
SET^1	Time for resource cluster changing from an initial status to an active in use status
SET^2	Time for resource cluster changing from active to the initial inactive status
n_e	Number of workstations
$\mathbf{H}_{ye}$	Setup time matrix
$(\xi_{ye})_{hk}$	Setup time
t_{ij}^{SET}	Optimal setup time
λ_{ixp}	Indicates whether part *i* employs assembly process *p* in plan *x*
η_p	Indicates whether process *p* is used in plan *x*
T^{SET}	Total setup time of all the parts
T^p	Total process time of all the parts
D_i	Volume of demand for part *i*
t_{ij}^p	Process time
$(a_{ye})_{hk}$	Indicates whether $(\xi_{ye})_{hk}$ is a finite non-zero number
L^p	Part priority
L^m	Assembly method priority
K	Utilization rate coefficient
τ_{ij}	Assembly time
γ_{ij}	Indicates whether part *i* process *j* is allocated in current workstation
Γ	Exponential recovery function
E	Resource utilization index
$\mathbf{B}_{ye}$	Binary matrix to calculate setup time

Chapter 8

$CC(i)$	Coordination cost of process *i*

size	Size of DSM
$n_c(k)$	Number of processes contained in cluster *k*
p	The penalty assigned to the size of cluster
DSM(i, j)	Value of the interaction between process *i* and *j*
$w(i, j)$	Weight rating of process flow from *i* to *j*

Chapter 10

CI	Process commonality index
x	Process plan
y	Resource allocation plan
X	Process plan set
Y	Resource allocation plan set
n_p	Total number of processes
n_d	Demands of product
u	Number of feasible resources
C	Process flexibility index
F	Process sequencing flexibility index
A	Process feasibility index
n_e	Number of workstations
$(\xi_{ye})_{hk}$	Setup time
t_{ij}^{SET}	Optimal setup time
λ_{ixp}	Indicates whether part *i* employs assembly process *p* in plan *x*
η_p	Indicates whether process *p* is used in plan *x*
T^{SET}	Total setup time of all the parts
T^p	Total process time of all the parts
D_i	Volume of demand for part *i*
t_{ij}^p	Process time
$(a_{ye})_{hk}$	Indicates whether $(\xi_{ye})_{hk}$ is a finite non-zero number

L^p	Part priority
L^m	Assembly method priority
K	Utilization rate coefficient
E	Resource utilization index
$W(i, j)$	Weight rating of process flow from i to j

ACO	Ant Colony Optimization
AHP	Analytical Hierarchy Process
ALB	Assembly Line Balancing problem
AMHS	Automated Material Handling System
ANOVA	Analysis of Variance
ASPEN	Assembly Sequence Planning and Evaluation system
BOM	Bill of Material
CAD	Computer-Aided Design
DBD	Dimensionless Block Diagram
DEA	Data Envelopment Analysis
DES	Discrete Event Simulation
DOE	Design of Experiments
DSM	Design Structure Matrix
ERP	Enterprise Resource Planning
GA	Genetic Algorithm
GBOM	Generic Bill of Material
GBT	Graph Based Theory
GPcS	Generic Process Structure
GPdS	Generic Product Structure
GPPS	Generic Product and Process Structure
GUI	Graphical User Interface
KKT condition	Karush–Kuhn–Tucker condition
LFGA	Leader-follower Genetic Algorithm
LLC	Lower-level Chromosome
LP	Linear Programming
MM	Mathematical Modeling
MPEC	Mathematical Program with Equilibrium Constraints
MTM	Methods Time Measurement
NN	Neural Network
NP-hard	Non-deterministic Polynomial-time hard
NVA	Non Value Added
OEE	Overall Equipment Effectiveness
P/D	pick-up/drop-off
PEM	Pairwise Exchange Method
PFA	Product Family Architecture
PLC	Programmable Logic Controller
PSO	Particle Swarm Optimization
QFD	Quality Function Deployment
REL chart	Relationship chart
RFID	Radio-frequency Identification
RWS	Roulette Wheel Selection

SA	Simulated Annealing
SLP	Systematic Layout Planning
SysML	System Modeling Language
TCR	Total Closeness Rating
TDB	Transaction Database
TS	Tabu Search
ULC	Upper-level Chromosome
UML	Unified Modeling Language
VBA	Visual Basic for Applications
VSM	Value Stream Mapping
WCM	World Class Manufacturing
WIP	Work in Process
XMI	XML Metadata Interchange

SUMMARY

Continuous growing demands of customized products, increasing competition in manufacturing industries and increasing labor cost are demanding mass customization to be realized in all spectrum of industry. However, mass customization has not lived up to its promise. The assembly system, being identified as the breaking point to enable mass customization, also brings challenges when dealing with high variety products which is typical situation in mass customization. The topic of this dissertation is identified as *game theoretic optimization of high variety assembly system design*. It suggests itself as a key enabler of mass customization paradigm, which allows companies to supply high variety product for today's market that demands customization without too much tradeoff in cost, quality or distribution. The research problem is formulated as to provide a systematical approach to design a multi-stage, multi-product high variety assembly system under multiple pre and post-conditions and constraints.

The proposed work is geared towards a game theory based solution to solve complex engineering system design problems met in mass customization, and using the high variety assembly system design as a demonstration. The dissertation reveals the fundamental issues underlying high variety assembly system design and decision making, which represents a typical complex engineering system. In order to tackle the fundamental issues, a technical framework of game theoretic optimization of high variety assembly system design is proposed. Accordingly, mathematical and computational models are developed within the framework to support 1) variety propagation from product to assembly process, 2) assembly system layout design, 3) assembly process

design and resource allocation, and 4) assembly process planning. These coherent models along the technical framework lay the theoretical foundation of this research, as described below.

First, in order to identify the necessary process elements and their relations for given product variety demands, the mapping from the product variety to process variety must be formulated. By using a generic representation method of both product and process information, the large amount of variety data of both product and process can be handled. Then the construction of association rules mining makes it possible to find a suitable assembly process set to deliver the process variety which fulfills the product variety demands.

Second, the major decision making problem underlying the high variety assembly system design problem is to find the equilibrium solution between assembly process design and resource allocation. With the evaluation criteria of assembly flexibility and resource utilization rate, the assembly system design decision making problem becomes leverage between flexibility and efficiency of the system design. The game theoretic optimization framework together with modified genetic algorithm brings a mathematical solution to this problem.

Third, the layout design of assembly system is the foundation of the of high variety assembly system realization. The task of designing an assembly system layout can be generally concluded as grouping similar and highly dependent assembly processes together into an enclosed unit, a department or a work station, depending on their scale. The use of design structure matrix enables the given assembly process set to be translated into a system layout which will optimize the material flow efficiency and minimize possible bottlenecks in the system.

Fourth, as a solution to high variety assembly system process planning and verification / validation of the assembly system design, a real-time data driven simulation method and an industrial application case study is reported. The data driven simulation brings not only a methodology to verify the system design, but also a potential industrial application with online simulation and decision making capability to keep improvement of the running assembly system. The case study as a validation to the methodology set proposed in assembly system design, illustrates the process flow to solve a real life high variety assembly system design problem.

CHAPTER 1

INTRODUCTION

This chapter provides an overview of background which leads to this research topic. Through the discussion of research motivation, the topic of research is identified as *game theoretic optimization of high variety assembly system design*. It suggests itself as a key enabler of mass customization paradigm, which should allow companies to adopt mass customization strategy in their assembly systems. Accordingly, the research objectives and scopes are defined, along with a technical roadmap outline for this research.

1.1 Research Motivation

In the last several decades, the increased global competition from the emerging economies in developing nations, and the increasingly fickle consumer looking for variation and individualized products, had led many researchers and companies to agree that an economic motivation for mass customization exists (Pine, 2000). Especially in high-wage countries, there is a clear trend toward standing out from global competitors by means of product differentiation. The cost leadership strategy, which aims to provide products that can only be manufactured cheaply in mass production, is no longer sufficient to keep their advantage in global competition. By adopting differentiation strategy, in which giving customization options for products is one of the major tactics, company will provide products that can offer unique attributes that are valued by customers and that customers perceive to be better than or different from the products of competitors. However, ever shorter product life cycles, growing number of product variety demands and lack of skilled workers in combination with increasing wage costs force companies to no longer follow only one strategy. Instead of commitment to either differentiation or cost leadership strategy, the advantages of both strategies have to be combined in order to offer customized products at competitive prices. As a result, many companies choose a mass customization strategy to reconcile individualized products with advantages of the economies of scale (Müller et al., 2013). As a production paradigm, mass customization allows companies to meet the customization demands by manufacturing relatively high variety product without too much tradeoff in cost, quality, or market size.

The concept of mass customization was put forth nearly 25 years ago. Yet, despite great strides in information technology, engineering design practice and manufacturing technology, which are the components necessary to make the paradigm realizable, mass customization has largely not lived up to its promise. Few examples of successful mass customization implementations, such as personal computers, are largely limited to certain systems where the existing, dominant product architecture enables mass customization to be viable. On the other hand, modern industrial shop floors are highly affected by the ever-increasing product variety and volatile market demands introduced by the mass customization paradigm. The task of design, planning and operation of manufacturing systems is becoming more and more challenging for companies, as globalization, mass customization and the turbulent economic landscape create demand volatility, uncertainties and high complexity (Mourtzis et al., 2014).

The expansion of the mass customization paradigm is dependent on developing methods and tools that support designers throughout the product development process. It will be critical to overcome the challenges brought by high variety of product which any company considering mass customization would have to work through. Variety can be achieved at different stages of product realization, during design, fabrication, assembly, at the stage of sales, or through adjustment during the usage phase (Hu et al., 2011).

Variety implemented in design stage incorporates customer design inputs. These kinds of products are personalized products and usually single piece order. Variety can also be achieved in the fabrication stage, by using different manufacturing methods, parameters and materials. For example, optical lens, fasteners, jewelries and many medical products are customized in fabrication stage. As they normally share the same design platform, the variety is limited comparing with design stage customization. Assembly is one of the most cost effective approaches to achieve high product variety. With a proper design, each functional module / sub-assembly of the product is provided with several variants so that the total assembly combination will provide high variety in the final products (Hu et al., 2011). Variety can also be added at the stage of sales. For example, golf clubs and trousers are often cut to the length at the time of purchase based on the user's height, waist watch band length can also be adjusted based on user's waist size. Some product can also be adjusted after purchase, such as some tennis and badminton rackets which have different weight add-ons. The products which can add variety at or after sales are generally mass produced products which can be manufactured in high volume with only one or several models. However, their varieties are always limited and only certain kinds of products can be customized this way. As shown in

Figure 1.1, five different typical manufacturing systems are shown. Variety achieved in design, manufacturing, assembly, sales and after sales can be substantially corresponded to these systems.

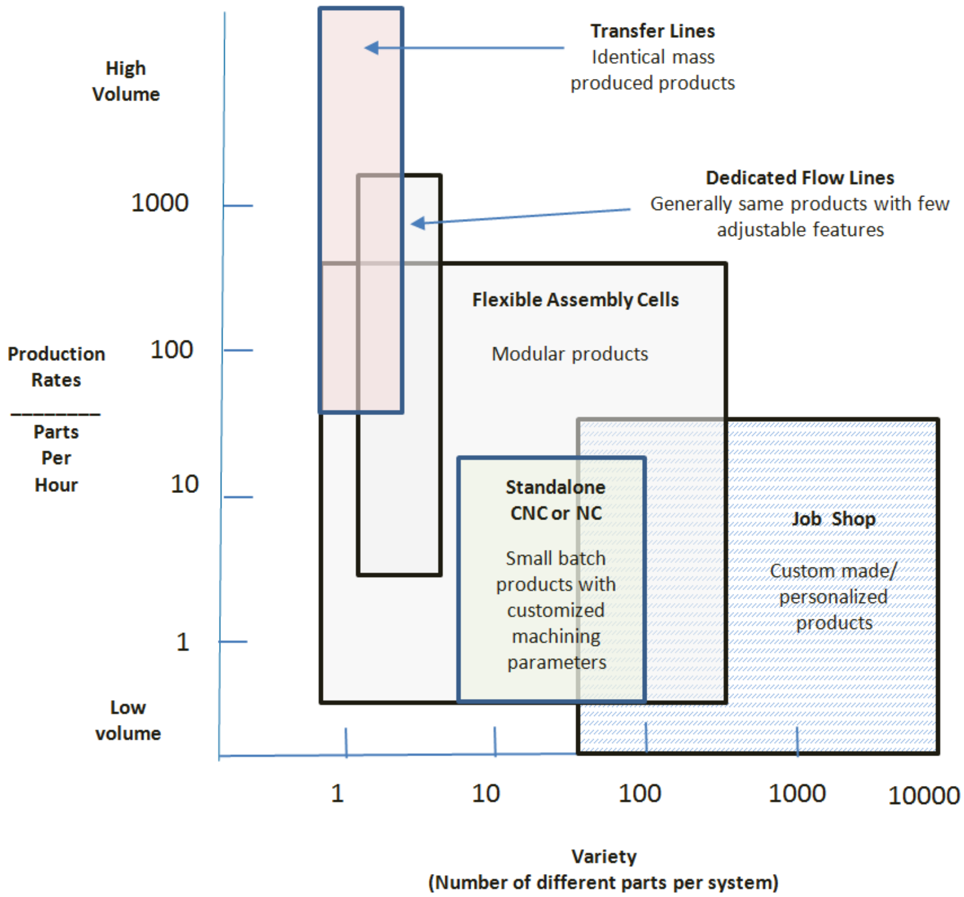


Figure 1.1 A comparison of different kinds of manufacturing systems

Assembly is the capstone of product realization process. It is also the key stage to add variety to products. From the comparison, it is clear that realizing high variety in assembly process is the most promising method in terms of balance between variety and economies of scale. It is also implementable for most of the manufactured products whereas other stages of variety realization have more limitations on product types. With development of robotic and machine vision technologies, many of today's manufacturing companies already adopted automatic or semiautomatic robotic assembly systems.

Different from the traditional transfer lines which majorly use delicate tooling, fixtures and automatic machines to produce certain predefined products, robotic assembly systems with help of intelligent machine vision, flexible feeding and material handling systems can easily handle multiple products or even multiple product families' assembly tasks. These developments in flexible manufacturing technology bring huge potentials on high variety assembly systems development.

1.2 Research Objective

Traditional assembly systems are designed to work with very limited kinds of products, the changeover time form one product to another will also significantly affect the total system efficiency. In order to design an assembly system which will be able to handle high variety products, there are several key technical challenges and corresponding research tasks can be identified:

(1) Variety modeling and propagation: High variety products implemented in assembly stage leads to high variety in assembling processes. In order to manage variety propagation from design to production, it is logical to model the variety in products, processes and their correlations. The related research tasks are:

a. Identify a suitable representation format for both product and process variety;
b. Propose a solution to extract product and process variety information form common company's product database;
c. Analyze the relationship of product and process variety;
d. Develop a mapping method to find suitable process variety information from given product variety information.

(2) Process modularization: The key enabler towards achieving customization in assembly process is product modularity and postponement strategies. The division of a product into separate modules / components provides the means to achieve high product variety at lower costs. Modularity, as a variety enabler, need to be applied in both assembly process level and assembly system level. The related research tasks are:

a. Find key factors of a modularization decision that effects the quality of high variety assembly system design;
b. Identify evaluation criteria to quantify and evaluate the key factors of a modular design;
c. Find the solution method to identify the optimal assembly process and system modular design.

(3) Assembly system configuration: Assembly system normally consist multiple assembly units, machines or setups. To carry out the high variety assembly processes efficiently with minimal system changes is a key to achieve a desirable production rate, thus keeps low average production cost. The related research tasks are:

a. Identify the assembly system configuration representation method which is suitable for optimization;
b. Identify evaluation criteria to quantify and evaluate the quality of assembly system configuration;
c. Find the solution method to identify the optimal assembly system configuration.

(4) Assembly operation planning: The consequence of high product variety manifests itself through an exponentially increased number of process variants, which

introduces significant constraints to production planning and control. The related research tasks are:

a. Identify the key factors that in a factory floor operation planning that effects the performance of the whole system;
b. Propose a method to enhance the system performance under the high variety production scenario;
c. Find the solution to identify the optimal process plan for factories dealing with high variety products.

(5) Optimization for equilibrium solution: In high variety assembly system design, there are always multiple criteria which are conflicting with each other, which is common in such a complex engineered system. There is no ultimate solution which will provide optimal results meet all the parties' interest at one time. A joint optimization framework is needed to reflect the conflict nature of this kind of decision making problems as well as providing an equilibrium solution. The related research tasks are:

a. Develop a optimization framework that will address the conflict nature of assembly system design decision making problems;
b. Propose a solution strategy to solve joint optimization problem;
c. Formulate a solution algorithm suitable for high variety assembly system design application.

1.3 Research Scope

The game theoretic optimization for high variety assembly system design is proposed as a new paradigm to approach the realization of mass customization by introducing game theory based decision making solution. It attempts to bridge the gap between the already well developed mass customization theories and manufacturing enterprise practice. As shown in Figure 1.2, first, the research is motivated by the current merging business strategy of cost leadership and differentiation, the volatile market demands together with increasing global competition require companies to produce customized products at low cost; Then the scope narrows down to identify the realization of mass customization in manufacturing floor, which is the major challenge in this field of research; Then in the third step, the key to mass customization realization is identified as high variety assembly system. The traditional all-in-one optimization methods are no longer suitable for such multi-objective optimization problems with equilibrium solutions. A game theoretic optimization with modified genetic algorithm solution is proposed to solve such problems; In the fourth step, from the design aspect of a product family realization, an assembly process design framework which enables high product variety through minimal tradeoff of cost and efficiency is formulated. This is the key enabler of achieving high variety in a given product family design; Following in the manufacturing phase, the assembly system design and layout optimization which is suitable for high variety production is proposed; In the sixth step, after all related design tasks in product level and factory level are studied, the high variety assembly process planning method in a factory floor setup is developed. Then as a way to validate the high variety assembly system design, a case study is conducted.

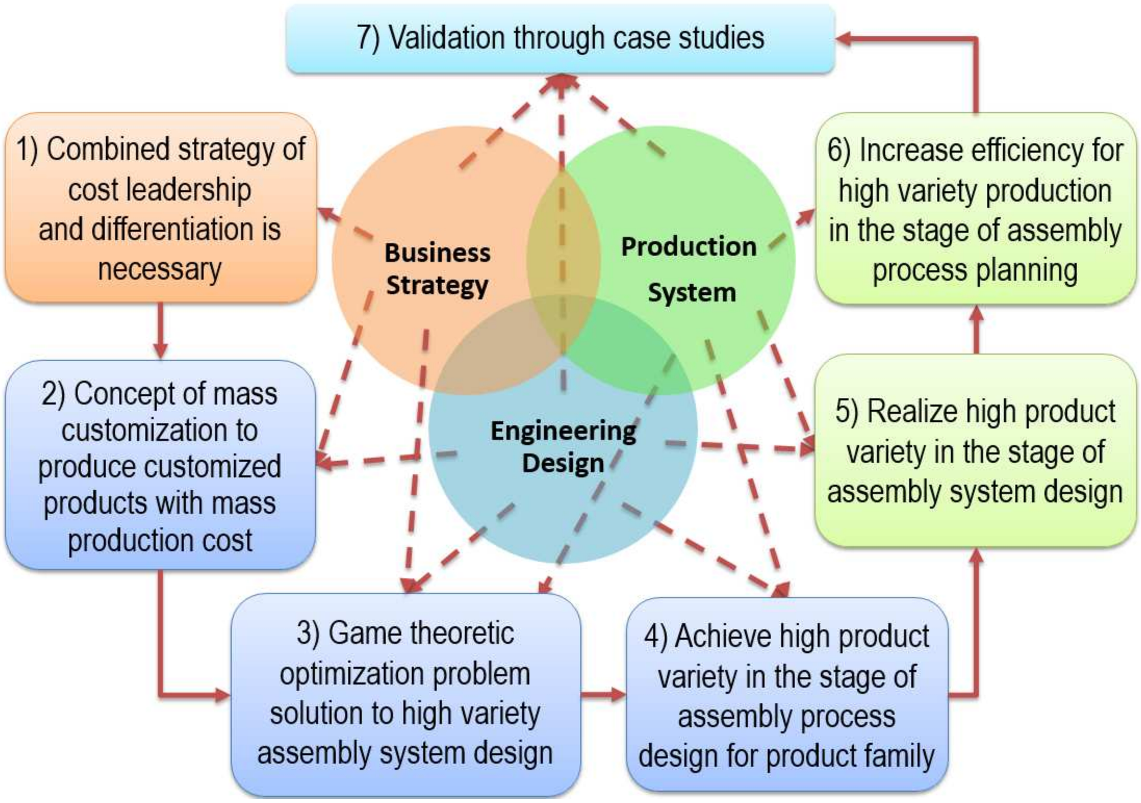


Figure 1.2 Research scope and research methodology

Thus, the proposed research spans over the intersection of game theory based design decision making, assembly process design, assembly system design and process planning by integrating fundamental principles from multiple disciplines across domains of engineering design, business strategy, and production systems.

1.4 Organization of This Dissertation

Figure 1.3 presents the technical roadmap of this dissertation, including motivation & significance, problem formulation, technical approach, methodology & solution, and validation & application.

Chapter 1 discusses motivation and significance of this research topic, along with a holistic view of research goals and scope. Chapter 2 provides a comprehensive review of various topics related to this research.

Chapter 3 formulates the key problems of this research. It presents the fundamental issues underlying high variety assembly system design. These fundamental issues help providing insights into how to solve them systematically.

Chapter 4 and 5 propose a technical solution framework for the high variety assembly system design problems. Chapter 4 introduces the game theoretic optimization method and its formulation. Chapter 5 proposes a leader-follower genetic algorithm used to search for optimal solution of game theoretic optimization problems.

Chapter 6 to 9 reports the development of high variety assembly system design starting from process variety propagation to assembly system process planning. Chapter 6 introduces a generic representation of product and process. Based on this generic representation, a product variety propagation schema is proposed. Then a rule mining application is introduced to find the relationship of product and process variety in product and process database. This rule mining result is used to generate a mapping between product and process variety thus provides propagation method to deliver process variety items from any given product data. Chapter 7 reports the derivation of different assembly process design evaluation criteria which is used to quantify the quality of a given assembly process design. These evaluation criteria are used as fitness functions in assembly process design optimization. Chapter 8 introduces a design structure matrix based high variety assembly system layout design method, which would deliver a clustering decision based on given assembly process set. Chapter 9 introduces a data driven architecture to formulate an online assembly process planning framework which could conduct real-time data acquisition and feedback to support assembly process planning decision making.

Chapter 10 presents a case study of the development of high variety car connector flexible assembly system. This case study is used to validate the high variety assembly system design framework and game theoretic optimization solution.

Chapter 11 concludes the thesis along with discussions of research limitations and future works.

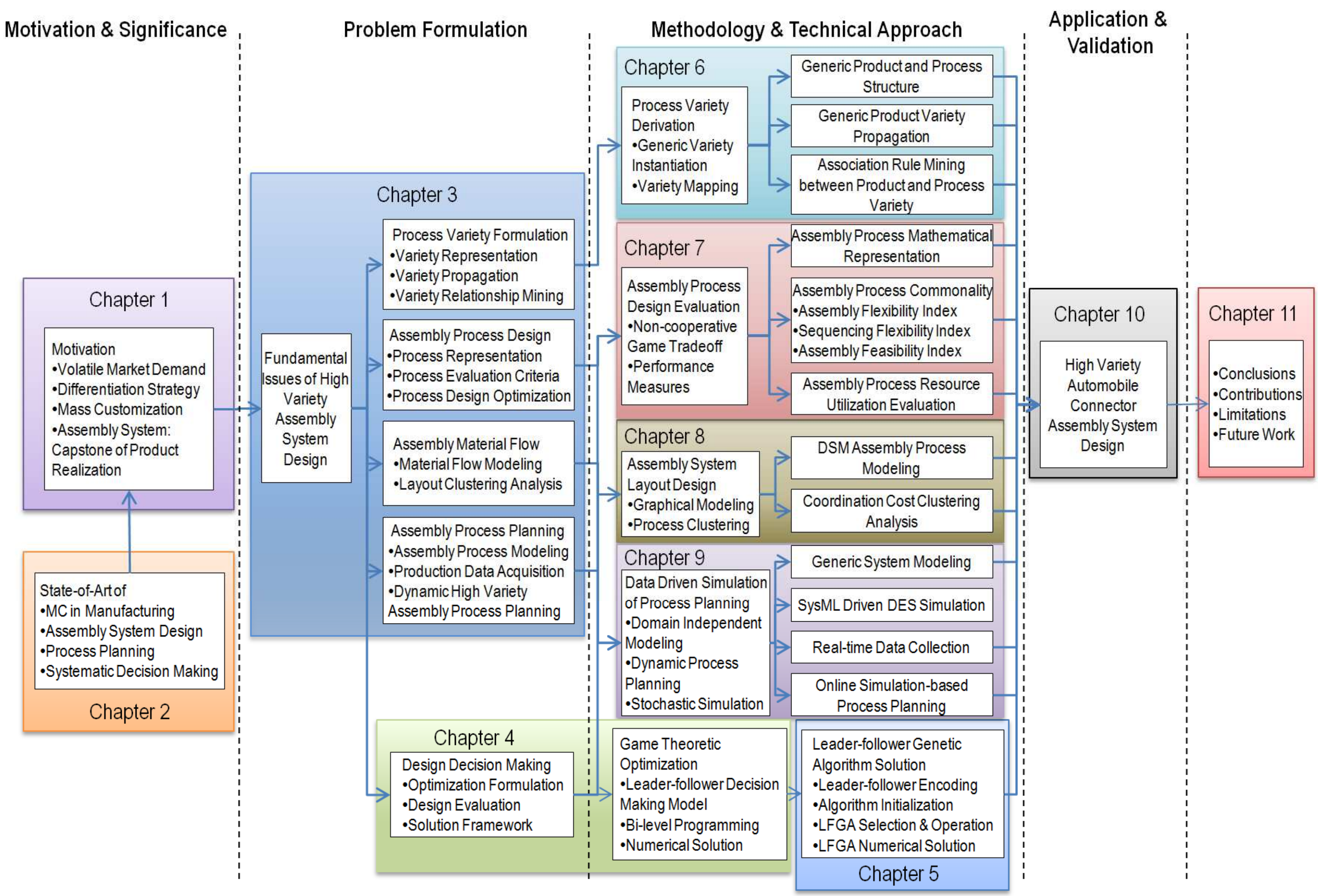


Figure 1.3 Organization of this dissertation

CHAPTER 2

LITERATURE REVIEW

This chapter is dedicated to the state-of-the-art review for game theoretic optimization of high variety assembly system design. Based on the research scope in Chapter 1, I will review various topics related to this research in mass customization applications, assembly system design, assembly process planning, and design decision making. A framework of reference will be given first that elaborates the topic relationships among different research domains. The limitations of the reviewed topics will also be discussed, which lead to the proposed methods in different chapters in the following.

2.1 Framework of Reference

As shown in Figure 2.1, this research mainly spans three domains, including business strategy, engineering design, and production system. In the domain of business strategy, the major topic is mass customization related business strategies. Both strategies in design phase and production phase are reviewed. Engineering design aims to build a product with a specified performance goal. It usually has a multi-step process, including task clarification, conceptual design, embodiment design, and detail design (Pahl et al., 2007). In this research, most related efforts in engineering design focus on process representation and design decision making in engineering systems. So the different representation methods of assembly systems and decision making approaches in complex engineered systems are reviewed below. Then in the domain of production system, topics related assembly systems design and planning are discussed, including assembly process sequencing, system layout design and assembly line balancing.

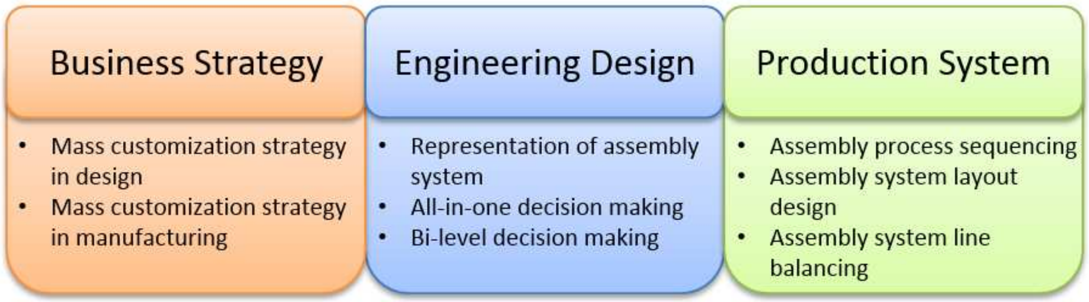


Figure 2.1 Various topics reviewed and their corresponding domains

2.2 Mass Customization in Manufacturing Industry

Literatures on mass customization related to manufacturing industries are commonly presented in two categories: product design definitions to facilitate mass customization and related manufacturing techniques.

Most of the literatures are focused on product design phase. Hölttä-Otto (2008), Dai (2007) and Du (2003) and their team proposed methods of modular identification to find out products internal functional relationships and physical properties’ similarities, which can support the possible customization opportunities. Fellini et al. (2002) and Jiao et al. (2000) also considered designer defined performance of product family into the final variant design. Yeh and Wu (2005) use module desirability and cost as determinant of the product modularization. Williams et al. (2007) integrates customer demand, range of variety and analysis of demand into a single problem formulation. After product modules being identified, the configuration of modules becomes a challenge to avoid “mass confusion”. Chen et al. (2010), Yang et al. (2009) and Huang et al. (2008) explore the development of product configuration method on rule-based, model-based and case-based constraints. Although most of the literatures choose product modularization as the answer to mass customization enabled product design, there are also some researches focusing on other solutions. Such as, Dai and Scott (2004) propose methods for identifying scalable components to handle the design challenges in mass customization. As product modularization is capable of generating product variety through the addition, substitution and exclusions of modules, it is the most commonly used method in product design phase in mass customization. However, the majority of publication is emphasized on product itself without considering the process modularity which is also very important in product realization in mass customization.

In the phase of manufacturing, mass-customized product often has higher or additional requirements than a mass-produced product because of the increased variety offered and the act of integrating the customer into the process of defining the final design configuration (Ferguson, 2014). The related works in this area have two distinct directions: production information management and new manufacturing technologies. Zhang and Efsthathiou (2006); Du and Jiao (2005); Jiao et al. (2000) and Tseng et al. (2005) have papers on generation of Bill of Material (BOM) and routing information to make production more efficient. Product and customer data management and the management structure are discussed by Zhao and Fan (2007); Waller (2004); Fan and Huang (2007) and Wang (2009). Besides those works in production information management, some manufacturing techniques such as selective laser melting by Vandenbroucke and Kruth (2007), combining reconfigurable molds and CNC machining by Kelkar and Koc (2008), reconfigurable robotic systems by Bi et al. (2004) and Zangiacomi et al. (2004) and rapid prototyping systems by Bateman and Cheng (2002).

Based on the review, we can see that many of the literatures are investigating the currently evolving additive manufacturing techniques, which is yet to be widely applied in industry. However, the traditional assembly system is still the backbone of manufacturing industry, which needs more attention to allow a smooth transition from mass production to mass customization.

2.3 Assembly Representation and Sequence

2.3.1 Assembly Representation

The first step in assembly system design is a process of analyzing the input product information, both geometric and non-geometric, to obtain the necessary assembly process information so that the subsequent assembly planning task can work. Such information is used to represent the assembly components and hierarchy, and to generate the sequences of assembly.

Many researchers have proposed graph-based assembly representational schemes, such as location graph (Eastman, 1981), virtual link (Lee and Gossard, 1985), constraint graph (Wolter, 1991), relational model graph (De Mello and Scaramelli, 1989), feature mating operation graph (Huang and Lee, 1989), functional relationship graph (Roy and Liu, 1988), and part position and part relation network (Heemskerk and Van Luttervelt, 1989). The basic concept is to store assembly entities, either parts, subassemblies, or parts with assembly operations, as vertices in various types of graph. The variety of relationships between assembly entities, such as connectivity, geometry, location, and functionality, are characterized in terms of joining edges between graph vertices. The more generic and commonly used assembly representation method in industry is the

BOM. A BOM generally lists all parts, subassemblies and materials, and also includes other information such as quantities, costs and manufacturing methods. A BOM usually has a tree-graph or tabular structure with hierarchical level codes (Hopp and Spearman, 2011). It has been a standard communication tool in industry for design, manufacturing and purchasing, and has been integrated to Computer-Aided Design (CAD) and Enterprise Resource Planning (ERP) systems.

The increasing product variety has led to new approaches in assembly representation. They appeared in the literature in diverse forms. Among these, the Product Family Architecture (PFA) has been one of the extensively studied topics. A PFA was used to measure market position, commonality and manufacturing economy (Tseng et al., 1996; Jiao et al., 1998; Jiao and Tseng, 1999; Jiao and Tseng, 2000; Jiao et al., 2007). There are also researches aiming at evolve BOM to represent a variety of products and processes, such as the concept of Generic Bill of Material (GBOM) (Jiao et al., 2000; Hegge and Wortmann, 1991; Olsen, 1997). These GBOMs use functional and structural relations among components to represent product variants. A variety of representation methods were used including tabular forms and programming language based notations. Other hierarchical representations were also used to represent product families. For instance, generic subassemblies for a product family were used for integrated product family and assembly system design (De Lit et al., 2003). Liaison graphs have also been adapted to represent product variety. One such development is the product family liaison graph that combines the liaison graphs of product variants by representing common components over different variants as a single node. Thus, for a family of products, the liaison graph can be modified to include both common and variant parts in the assembly. A product family liaison graph was used to identify maximal common subassemblies and a product-family assembly sequence (Gupta and Krishnan, 1998).

The current assembly representations are limited in terms of the comprehensiveness of assembly information. For example, the usual BOM cannot directly represent the complex physical assembly processes. On the other hand, the assembly representations based on the liaison graphs are not suitable in representing hierarchical functional structures. A new graph-theoretic assembly representation incorporating product and process information is necessary to overcome the above problems.

2.3.2 Assembly Sequence

The sequence of assembling parts and components plays a key role in determining the quality of the assembled product, as well as assembly process efficiency and complexity. Determination of all possible assembly sequences is an important and critical stage in the total design process of a assembly system design.

Bourjault's work (1984) used rules that are determined by a series of "yes" or "no" questions, which are answered by studying the mating of components for an assembly. De Fazio and Whitney (1987) extended Bourjault's work by simplifying the determination of the set of rules, or precedence constraints, by using specific questions about liaison precedence. Other work that takes advantage of a computer aid in determining all assembly sequences is the work of Khosla and Mattikali (1989). Kanai et al. (1996) developed a computer aided Assembly Sequence Planning and Evaluation system (ASPEN) that takes all the solid-model components of a product and automatically determines all feasible sequences by decomposition and determines the optimum sequence using Methods Time Measurement (MTM) as time standards for operating time determination. Choi and Zha (1998) developed computer aided automatic assembly sequence generation with their work on automated sequence planning. Mello and Sanderson (1991) built upon previous research by treating an assembly sequence generation problem as a disassembly sequence problem. Dini et al. (1999) made use of the genetic algorithm to create and evaluate assembly sequences. They created a fitness function which takes into account geometrical constraints of the assembly and other optimization aspects and using their genetic algorithm decreased the time for computation. Almost all assembly sequence generation algorithms are based on sequential tasks. Consideration of assembly hierarchy allows parallel assembly sequence and hybrid system configurations and such choices can be explored to simplify assembly sequence generation and system design (Li et al., 2011).

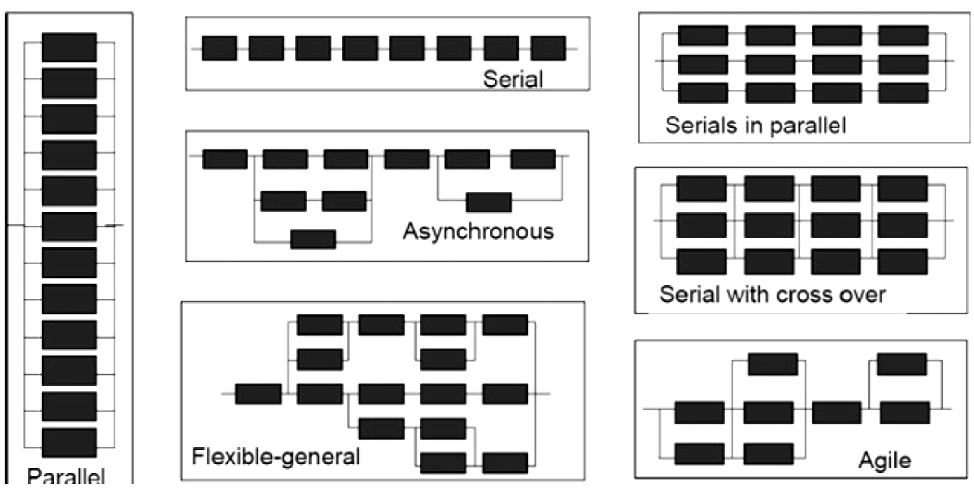


Figure 2.2 Example of different assembly sequences (Hu et al., 2011)

2.4 Assembly System Design and Balancing

2.4.1 Assembly System Layout Design

Assembly systems can be designed using various layouts. The moving assembly line introduced by Ford (2007) had a serial layout. Such systems, known as serial lines or flow lines, were used for high volume production of a single product type with dedicated machines and material handling systems. To achieve high variety in an assembly system, an efficient layout arrangement and material flow path design are important due to the large percentage of product cost that is related to material handling (Yang et al., 2005). A poor layout and flow path design can result in high material handling costs, excessive work-in-process inventories, and low or unbalanced equipment utilization (Heragu, 2008). Luggen (1991) defined four basic layout configurations: spine, circular, ladder, and open-field. A modern assembly system is usually equipped with an Automated Material Handling System (AMHS) and computer numerically controlled machines.

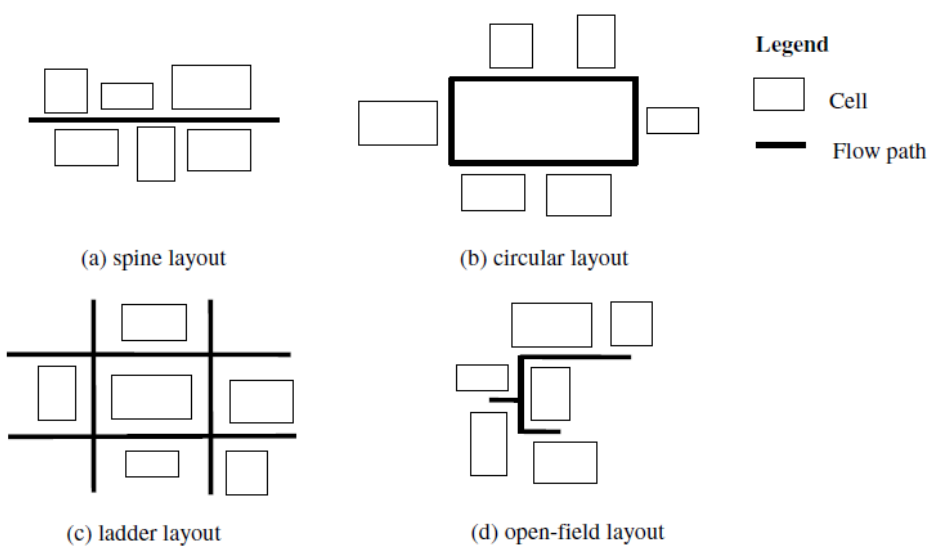


Figure 2.3 Types of assembly line layouts (Yang et al., 2005)

Montreuil and Venkatadri (1990) consider the dynamic manufacturing system and provide an interpolative design approach. This work did not force the cell shape as a hard constraint. In addition, the distance measure used is not directional. Montreuil (1991) proposed a modeling framework for integrating layout and flow network designs. Chhajed et al. (1992) provided a detailed flow network design on an existing layout. Given a fixed single-loop material flow path, Wu and Egbelu (1994) developed a procedure to determine an optimal layout design along this path. For this approach, the flow path is known a priori. The decision is to determine the flow sequence along this path. Chittratanawat and Noble (1999) proposed an integrated approach for facility layout, P/D location and material handling system design with equal-size department assumption. Banerjee and Zhou (1995) designed a directed, single-loop layout by sequentially determining the flow sequence between machines and the layout of the machines. Kim and Kim (2000) addressed an open-field type layout-planning problem for facilities with fixed shapes and P/D points.

2.4.2 Assembly System Line Balancing

In high variety assembly system, it will always need to handle multiple sets of work elements or tasks, each having a set of precedence relations. The assignment of tasks to an ordered sequence of stations in order to satisfy the precedence relations and optimize the effectiveness is commonly categorized as an Assembly Line Balancing problem (ALB). Performance optimization for ALB problems can be done by various methods namely, by Heuristic, Mathematical Modeling (MM), Design of Experiments (DOE), Hypothesis testing, Analysis of Variance (ANOVA), Analytical Hierarchy Process (AHP), Linear Programming (LP), Data Envelopment Analysis (DEA), Simulation approach, and also by search algorithms like Genetic Algorithm (GA), Simulated Annealing (SA), Neural Network (NN), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO) etc. Shift towards the use of GA is increased in recent years (Rane et al., 2015).

The most commonly used heuristic approach in ALB is lean manufacturing techniques, which introduces a pull system with a short lead time and aim to eliminate different kinds of wastes (Shah and Ward, 2003). The various lean techniques used are ALB, reduced Work in Process (WIP), minimizing Non Value Added (NVA) activities, use of kanban pull system, Value Stream Mapping (VSM), Quality Function Deployment (QFD), World Class Manufacturing (WCM) etc.

Mathematical model plays important role in defining mathematical relationship between input and output. It is an exact method of obtaining the solutions to the system under consideration. Adham and Tahar (2012) used LP model and GA to minimize the queuing problem, idle time and regulate the workers. Hu and Lin (2009) establish multistage scheduling model of an auto vehicle mixed model assembly line. The model is solved by Lingo software and the optimal scheduling of queue was obtained. For complex mathematical model, the concept of GA is used to generate processing sequence.

Simulation can clarify the exact nature of the tradeoff between customer satisfaction and cost-effective delivery of service. Venkat (2006) demonstrated the significant role that DES can play in design of cost effective system. Mixed model lines offers increased flexibility. Bottleneck management can be used for optimizing manual automobile assembly system. This was illustrated by Dewa and Chidzuu (2012).

2.5 Decision Making Approaches

Balancing of multiple engineering decision making is commonly achieved by integrating the different engineering domains as one singular optimization problem, such that multiple design criteria are aggregated into one “all-in-one” objective function (Luo, 2011). While multi-objective optimization approaches address standalone design problems well, the decision making in a complex engineered system is related to coupling of multiple decisions, which needs a synergy of conflicting goals of each individual engineering optimization problem. The all-in-one approach is often practically infeasible in such situations due to computational and organizational complexities (Alyaqout et al., 2011). Optimization by decomposition has been appealing for alleviating the problem of handling a large number of design variables and constraints simultaneously (Kokkolaras et al., 2006). Decomposed optimization largely works only if the domain problem follows a hierarchical decision flow. However, many problems (e.g., product portfolio planning) involving marketing-engineering concerns cannot hierarchically decomposed along disciplinary boundaries. Coordination between multiple engineering decision makers indeed implies equilibrium decisions, whereby different parties strive for different interests and have to compromise with others to achieve common solutions (Devendorf and Lewis, 2011). A bi-level programming is becoming more popular method among the engineering decision making approaches.

Bi-level programming, also known as bi-level optimization, refers to the mathematical programming model, whose constraints contain sub-optimization problems. It is first studied and proposed by Bracken and McGill (1973). It abstracts a class of hierarchical decision-making problems including the leader-follower decision-making problem. Owing to its wide practical relevance, the bi-level programming theory has become an important branch of mathematical programming. Nonetheless, it still has many characteristics of its own. Normally, the upper-level problem contains optimization functions or solutions of the lower level. As a result, the bi-level model itself is non-smooth. Moreover, even linear bi-level programming problems are Non-deterministic Polynomial-time hard (NP-hard) (Hansen and Savard, 1992; Vicente et al., 1994). When the upper-level function contains an optimal solution of the lower level, the feasible region could be disjunctive. Currently, there are several methods being used to solve bi-level programming problems, including the *K*-th optimal solution for special linear conditions (Bialas and Karwan, 1982, 1984), the branch-and-bound method (Edmunds and Bard, 1992), the method of replacing sub-problems by its Karush–Kuhn–Tucker condition (KKT condition) to form a single-level programming problem (Fortuny-Amat and McCarl, 1981), the method of using duality gap penalty function to form a single-level programming problem (Anandalingam and White, 1990), intelligent algorithms (Mathieu, 1994), etc.

Due to the theoretical difficulty and computational complexity in bi-level programming problems, they are rarely used in practical applications such as engineering design. Roy et al. (2008) have a systematic survey on optimization methods in engineering design with 202 references, but no bi-level optimization is included. With the development of engineering design, many important design issues have shown complex structures with a leader-follower hierarchical feature, which can hardly be tackled with traditional single-level optimization models. The bi-level optimization is more widely known from its increasingly important role. In recent years, there are some researchers investigating the bi-level programming problems in engineering design applications. For instance, Shabde and Hoo (2008) establish a chemical product design and process control joint optimization model based on the bi-level programming framework. Nicholls (1996) applies bi-level programming in aluminum production planning. Wang et al. (2008) use a bi-level programming model to solve workshop production scheduling problems.

2.6 Summary

The topics reviewed in this chapter offer guidance to solve the fundamental issues involved in high variety assembly system design in the next chapter. Considering the limitations of various topics reviewed here, I propose methodologies that can overcome their respective limitations in Chapters 4, 5, 6, 7, 8, and 9 to address a specific step of the high variety assembly system design and its game theoretic optimization solution.

CHAPTER 3

FUNDAMENTALS OF HIGH VARIETY ASSEMBLY SYSTEM DESIGN

Recognizing the importance of assembly process in high variety products, this chapter examines the fundamental issues underlying assembly system design for high variety products, which includes formulation and representation of process variety, assembly process evaluation, assembly system layout design and assembly production process planning. Understanding these fundamental issues is crucial to this research and each of them will be further discussed in later chapters with consideration of game theoretic optimization solutions.

3.1 A Holistic View

The key challenges for high product variety realized through assembly system can be viewed from two aspects: In product level, the focus is on the derivation of an assembly process design to allow efficient variety generation, which involves a certain product family design; In factory level, the focus turns to the realization of product assembly process to produce high variety products, which deals with one or more product families in a certain factory floor. By further dividing the high variety assembly system design issues with different phase of product realization, four contextual research topics can be identified. Each of them can be viewed as one part of the logical process from inputting a high variety product definition to delivering a functional high variety assembly system.

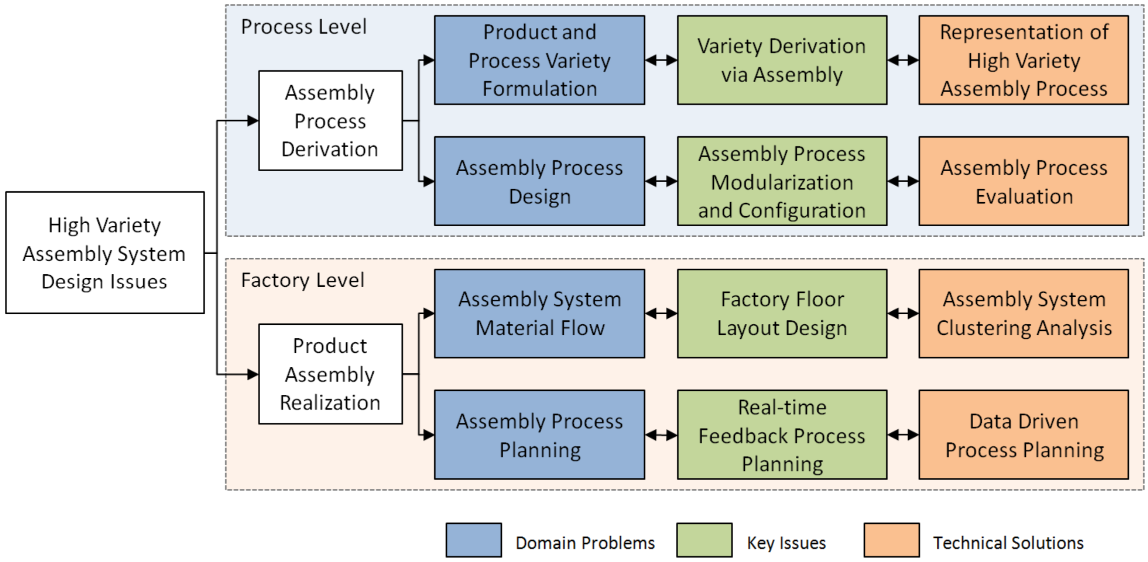


Figure 3.1 Fundamental issues involved in high variety assembly system design

3.2 Process Variety Formulation

The first step of high variety assembly system design is to identify the necessary process elements and their relations for a given product architecture, in another word, the mapping between the product variety to process variety.

To utilize commonality underlying product diversity and process variation, it has been widely accepted as a practice to develop product families, in which a set of similar variants share common product and process structures and variety differentiates within these common structures. Traditional BOM is an efficient way to represent product structures, but it needs unification and adoption to deal with variety. When it comes to the variety in assembly process, the major challenges lie in the relationships between diverse product variants and the corresponding production process variations as well as the selection of various operations alternatives with respect to a large number of functional requirements and their combinations (Jiao et al., 2000).

The assembly process should reflect the flow of material through the production process. By extending the BOM with adding variety parameters and assembly configuration constraints, it is possible to modify traditional BOM structure into a generic variety structure which provides a concise way to characterize variant derivation at different levels of the structure, variety parameters, and/or parameter values, for both product architecture and process flows.

The derivation of product variety also requires a reusable assembly process representation. In the phase of assembly process design and verification, there will be over thousands of combinations of different product variances coming from the same level of quantity of assembly process designs. It will be very time consuming for manually derives variances from each different assembly process, the practice of verification and evaluation would be even more time consuming. As a result, an assembly process representation which allows automatic generation and test that can be reused for all different products is key to the real life applications of high variety assembly.

A generic mathematical representation of both product variety and process variety is also very important. In the decision making process of assembly system design, the mathematical representation of each design variant, including information of both product variant and its process variant, will be a fundamental element for modularity quantification and optimization in the following study. The research problem in process variety formulation can be concluded as following:

a. Develop a concise and systematic variety representation for both product and process variety;

b. Formulate a variety propagation method from customer demand to product variety items;

c. Propose a method to identify the mapping relationship from product variety to process variety, which can be used to derive process variety on given product variety specification.

3.3 High Variety Assembly Process Design

Motivated towards delivering high variety at low cost, the idea of process modularization is to reuse common assembly processes and resources. Process similarity is the foundation of clustering analysis in its modularization. The objective of assembly process modular design is to group related processes into one module so each common process resource can reach higher utilization rate while simplifying the whole process flow into a modular system. On the other hand, the efficiency of assembly process is equally important as similarity. A good assembly process modularization plan should achieve efficient utilization of assembly resources as well as maintaining its flexibility to ensure a reasonable amount of varieties of products can be assembled with the same process design.

To deliver an optimal assembly process design in terms of both flexibility and efficiency is a challenging task without good evaluation methods. Given multiple schemes of assembly process design, an optimal alternative could be selected with respect to different sets of evaluation criteria. For example, an appropriate assembly process plan can be decided according to status of recourse utilization (Tonshoff et al., 1989), by shortest processing time (Kim and Egbelu, 1999; Jian et al., 2006; Leung et al., 2010), in light of minimal tardiness (Weintraub et al., 1999), in terms of maximal diversity of equipment engaged to fulfill all the job (Saygin and Kilic, 1999), or based on whichever achieves the lowest manufacturing cost (Wang et al., 2008; Haddadzade and Farahnakian, 2009). Based on the nature of high variety assembly systems for mass customization, proper evaluation criteria with a comprehensive effort of these factors will be further discussed in Chapter 7.

As the process flexibility and efficiency are competing goals in assembly process modularization, such decision making problem is not able to be described using traditional all-in-one solution. This kind of decision making problems will be addressed in the introduction of game theoretic optimization in Chapter 4. The research problem in high variety assembly process design can be concluded as following:

a. Identify a mathematical representation of process plan to allow quantification of assembly process design quality;
b. Formulate suitable evaluation criteria to quantify the quality of a given assembly process design;
c. Find a solution method to identify the assembly process design solution.

3.4 High Variety Assembly System Material Flow

Following the product and process representation formulation and design modularization, a suitable assembly process design for a given product family can be derived and evaluated. In product level, such information will represent a full description of how the product variety can be achieved in an assembly system. However, if we look at them in a production perspective, the product family design incorporating high variety and the corresponding assembly process is just one of many components and information that is merging into the factory floor. They can be considered as one of the inputs of assembly system design task.

One of the major issues in assembly system is the arrangement of the multi product and multi task material flow control and planning. Traditional assembly systems are lines of dedicated workstations in which parts are added as the semi-finished assembly moves from workstation to workstation. The parts are added in sequence until the final assembly is produced. Such assembly lines are widely used in industry since almost 100 years ago when the famous Ford Model T was mass produced. But such assembly systems are normally limited in terms of product variety and flexibility.

Modern assembly systems should have the ability to deal with slightly or greatly mixed parts, to allow variation in parts assembly and variations in process sequence, change the production volume and change the design of certain product being manufactured, especially in high variety assembly systems. Assembly system dealing with high variety and high productivity must be flexible and efficient in material flow path design. Taking these objectives as optimization goals and searching for an equilibrium solution, with consideration of physical constraints and stochastic demands is a NP-hard problem. To solve such problem, a heuristic clustering algorithm applied in design structure matrix will be formulated. The detailed discussion can be found in Chapter 8. The research problem in high variety assembly system material flow can be concluded as following:

a. Define a material flow modeling method to represent specific assembly process plan design;

b. Formulate an assembly process clustering analysis schema to identify the clustering trend of assembly processes in terms of best system efficiency and flexibility.

3.5 High Variety Assembly System Process Planning

Assembly process planning is an important function in production planning and control of discrete-part manufacturing. In factory level, the assembly process planning is not only concerned with determining the sequence of individual manufacturing operations needed to produce a given part or product, but also determining the factory equipment load balance and production schedule to ensure a efficient equipment utilization as well as meeting different customer demands. Planning and using efficient assembly processes across the whole factory floor can actively contribute to the reduction of a product's manufacturing cost. However, due to the complexity and the multiplicity of high variety assembly systems, the selection of the appropriate assembly process plan requires a high level of expertise and experience from the planner's side. Assembly process planning is a time-consuming procedure and, as a result, the automation of this procedure is necessary.

As product variants increase, variant-oriented planning of their assembly processes becomes an important logical enabler to support such a change on the assembly system. Such assembly system dealing with multiple products assembly at the same time is normally called a mixed model assembly system. It has been recognized as a major enabler to handle product variety. However, the assembly process becomes very complex when the number of product variants is high, which, in turn, may impact the system performance in terms of quality and productivity (Zhu et al., 2008).The performance of mixed model assembly system characteristics is dependent on product assembly process design, assembly system layout and the demand patterns that the production system is subjected to. In terms of automation solution of process planning for such system, the use of simulation is proving to be indispensable, since the NP-hard nature of the scheduling problem does not allow the identification of the optimal solution within an acceptable time frame. Toward this goal, Discrete Event Simulation (DES) with real time solution capabilities is being investigated in Chapter 9.

For products with long life cycles produced in large quantities and often assembled manually, time-consuming investments in process planning are justifiable. However, present market conditions and higher product variety lead to much shorter life cycles and smaller production volumes (Hu et al., 2011). On the other hand, the final production sequence is constructed step by step when operating the assembly process. Thus, it is not the result of a single decision but of a dynamic or rolling process. As a result, a new variety-oriented real time assembly planning is required, and corresponding real time data collection used for the online simulation and decision making becomes an essential part. The research problem in high variety assembly system process planning can be concluded as following:

a. Identify a modeling method to build simulation model of high variety assembly system which can perform process planning task;
b. Develop a system architecture which allows production data acquisition;
c. Propose a dynamic high variety assembly system process planning framework.

3.6 Summary

This chapter examines the fundamental issues underlying high variety assembly system design. These fundamental issues include product and process variety formulation, assembly process design for high variety products, assembly system layout design and simulation, as well as assembly system process planning. Their interrelationships and overall influence to the whole product life cycle of high variety products are also elaborated. Such a profound understanding of these fundamental issues provides us a clear direction for a technical approach in the next chapter. Research problems in each fundamental issues are also identified, which will be further addressed within the methodology and technical approaches in the following chapters.

CHAPTER 4

GAME THEORETIC OPTIMIZATION

4.1 Introduction

Engineering design decision making deals with various tradeoffs and constraints involved in meeting the goals of overall problem solving. Most enterprise-level planning and engineering-level design decisions are typically integrated as a single optimization problem that necessitates an all-in-one solution. Commonly multiple design criteria are aggregated as a single-level objective function, for example in the form of expected utility on profit, revenue, etc. (Hazelrigg, 1998). In practice, these kind of all-in-one approaches tend to be infeasible due to the computational and organizational complexities. While in many cases design decision making is enacted as one optimization problem with multiple decision criteria, certain decision scenarios comprise multiple optimization problems that are competing with one another and have to compromise to arrive at equilibrium optima, and each of the optimization problems itself may be associated with a different set of criteria. Such optimization of multiple competing optimization problems all together leads to a joint optimization problem. Joint optimization problems are frequently observed in complex engineered systems that involve diverse couplings of multiple sub-systems and typically a joint effort of sub-system optimization is required. This chapter presents a systematic formulation of a

Stackelberg game theoretic optimization model for high variety assembly system design, which is a typical complex decision making problem with competing criteria.

High variety assembly systems design involves multiple stages of design and optimization that requires decision making deals with different tradeoffs and constraints such as flexibility, commonality, utilization and throughput etc. It is very important to leverage these attributes to reach a solution that the system performance, product variety and cost can be optimized jointly. In order to deal with the joint optimization problems in different stages of high variety assembly system design, this chapter present a systematic formulation of a Stackelberg game theoretic optimization model for assembly system design evaluation.

4.2 Bi-level Optimization vs. All-in-one Approach

The most commonly adopted solution to deal with the optimization problems which have multiple disciplines is all-in-one optimization. A modern engineered system such as assembly system always contains more than one subsystem which leads to multiple optimization goals in their design optimization process. It is assumed that all subsystems can be aggregated to fulfill a general system optimization goal with common interest. Then the optimization procedure of multiple sub-problems is combined to one ultimate all-in-one optimization problem. The realization is normally weighted sum with weights pre-defined based on domain experts or calculated according to system sensitivity derivatives. The advantage of all-in-one solution is obvious, the aggregated problem is a single disciplinary optimization problem and there are plenty of mature algorithms can be applied to find the global optima.

However, the assumption of all-in-one solution that all the subsystems can be coordinated and having common interest is not always true. Actually it is assuming the design optimization is dealing with only one domain problem with multiple objectives. In high variety assembly system design optimization, there are multiple domain problems, such as assembly process flexibility and work station utilization, which is competing with one another. It is impossible and not reasonable to find an ultimate goal for all such optimization problems. The assembly process design problem in terms of assembly flexibility is trying to identify a set of assembly sequence and the manufacturing resource it uses in each sequence, so that the assembly process set can be used to assembly the product variants with minimal changeover effort in machines and tooling. On the other hand, the assembly process design in terms of resource allocation is trying to identify the ideal resource allocation plan which will be feasible for the given assembly sequence as well as maximize the utilization to all assembly machines. These two optimization problem are formulated in different domain of assembly process design and serving competing goals.

For a given assembly system factory floor resources, there are different types of machines and tooling which is capable of fulfilling multiple assembly tasks. And it is common that some of the newer machines are more flexible which have less impact to the total system changeover, while the other are less flexible which are faster for some delicate job but takes longer time to changeover. If just following the assembly flexibility goal, all assembly process should be done on the most advanced and flexible assembly machine, which will results a very low efficient assembly resource utilization, thus cause a high assembly cost and low productivity. However, if just following the resource utilization goal, some of the assembly process may suffers from very long changeover time due to using a non-flexible assembly process to produce high variety products with many changeover requirements.

Based on the traditional all-in-one approach, one can use utility theory and try to convert both assembly flexibility and resource utilization into the same utility measure and then solve both optimization problems at once. However, there are several reasons which make it infeasible in the case of high variety assembly system design in terms of assembly flexibility and resource allocation:

1) The feasible region and solution space of assembly process plan and resource allocation plan are dependent on each other, which means the solution for assembly process flexibility is constraint of available resource allocation plans and vise versa. And their correlation is implicit. As a result, it is infeasible to aggregate both optimization goal functions into one single utility function;
2) Fidelity of an all-in-one utility function to represent both assembly flexibility and resource utilization is hard to verify, and in many cases, it also suffers from computational and organizational complexities as they are optimization problems from different domain of assembly system.
3) The determination of assembly process plan in terms of assembly flexibility and resource allocation plan in terms of resource utilization are non-cooperate games, which will not have an optimal solution, tradeoffs must be made to reach an equilibrium solution. Such optimization of

multiple competing optimization problems all together leads to a bi-level optimization problem.

As a result, the decision making problem of high variety assembly system design in terms of assembly flexibility and resource utilization falls into the bi-level optimization category which one optimization problem is embedded (nested) within another.

4.3 Leader-follower Decision Making

In order to better represent the competing nature of the multiple disciplines in structural design optimization, a leader-follower joint optimization model can be used. It is originated from Stackelberg games (Von Stackelberg, 1952). A game theoretic optimization problem can be formulated as a two-level optimization problem between two decision makers. Each decision maker knows completely the objective functions and constraints of the other. The upper-level decision maker (leader) announces his decisions to the lower level (follower). And then the follower makes his specific decisions and feeds the decisions back to the leader. The basic form of leader-follower optimization coincides with the Stackelberg games. A Stackelberg game solution deals with the interplay of two self-interested decision makers who decide sequentially, implicating a mathematical program that contains sub-optimization problems as its constraints. In general case, the objective values mutually depend on the choices of the other party. Technically, the follower's role can be seen as solving a parametric optimization problem, whose parameters are determined by the leader. The Stackelberg model originated from strategic games in economics, it has been used to study sequential decision making problems in diverse fields. This obtained problem is a special case of Mathematical

Program with Equilibrium Constraints (MPEC), a terminology widely used in the literature nowadays. The problem of the design of multi-stage, multi-product high variety assembly system under multiple pre and post-conditions and constraints is investigated in this research. The problem is of NP-Hard computational complexity. To demonstrate that a problem is in the class of NP-Hard complexity, it is common practice to depict that it is at least as hard as another proven NP-hard problem (Garey and Johnson, 1979).

In a leader-follower optimization process, as shown in Figure 4.1, each decision maker adopts his own strategy to optimize its own payoff. In general case, the objective values mutually depend on the choices of the other party. Technically, the follower's role can be seen as solving a parametric optimization problem, whose parameters are determined by the leader. The followers and leader optimization alternately, producing a design improvement in each iteration. By starting from a best guess of initial design, the model improves design in iterative cycles, each cycle comprises two steps. In step one, the lower-level variables are frozen and the improvement is achieved by upper-level optimizations. In step two, further improvement is sought in the space of the lower-level variables. After enough iteration, an equilibrium solution will be reached.

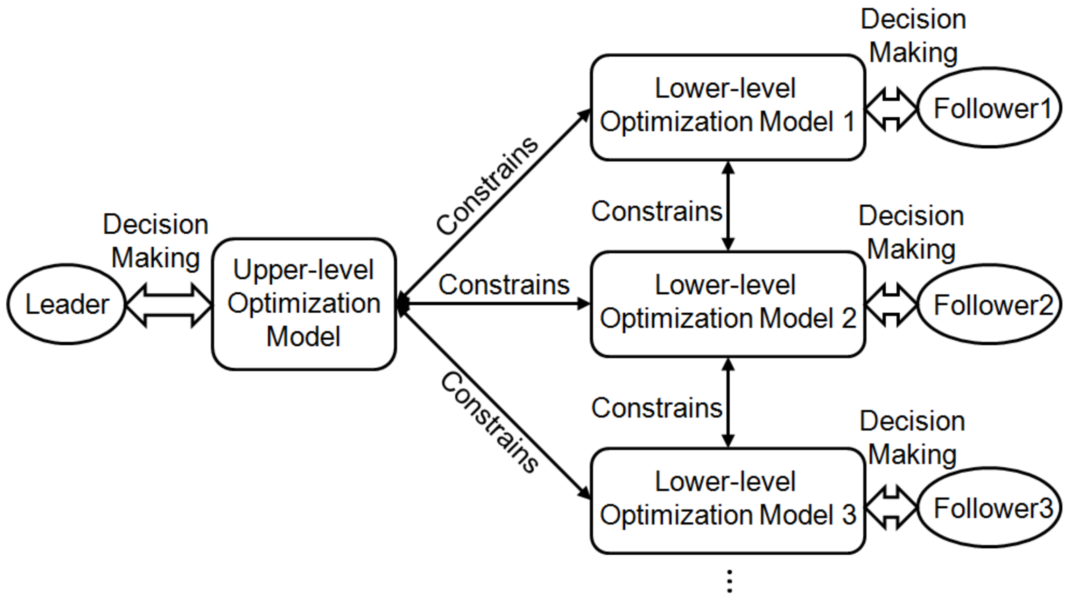


Figure 4.1 A leader-follower decision making process

This two-person game is introduced by Von Stackelberg (1952) in the context of unbalanced economic markets. In this model, the control of the decision variables is partitioned among the two players. Each player seeks to optimize his objective function. The leader goes first by choosing a vector in an attempt to optimize his objective function. In doing so, he must anticipate all possible responses of the follower. The follower observes a leader's decision and reacts by selecting a vector that optimizes his objective function. Because the set of feasible choices available to either player is interdependent, the leader's decision affects both the follower's objective and decision space and vice-versa.

4.4 Mathematical Formulation

Game theoretic optimization represents a framework for the analysis of joint decisions problems. A game theoretic optimization problem is described as finding a good decision without knowing exactly in which state the environment will be when this decision is implemented. The leader F_U has an *s*-dimensional design variable $\mathbf{X} \in R^s$. The follower F_L has a *t*-dimensional design variable $\mathbf{Y} \in R^t$. Then the game theoretic optimization formulation can be represented as follows:

$$Min_{\mathbf{X}} F_U(\mathbf{X}, \mathbf{Y}^*), \tag{4.1a}$$

$$\text{s.t.} \qquad G_U(\mathbf{X}, \mathbf{Y}^*) \leq 0, \mathbf{X} \in R^s, \tag{4.1b}$$

where$\mathbf{Y}^*$is an optimal solution of the leader's perception of the follower's decision on solution**Y**, and **Y** solves:

$$Min_{\mathbf{Y}} F_L(\mathbf{X}, \mathbf{Y}), \tag{4.1c}$$

$$\text{s.t.} \qquad G_L(\mathbf{X}, \mathbf{Y}) \leq 0, \mathbf{Y} \in R^t, \tag{4.1d}$$

where G_U and G_L are vector valued functions of dimension *p* and *q*, showing the constraints. From Eq. (4.1), the constraint region of the design variables can be denoted as $\Omega = \{(\mathbf{X}, \mathbf{Y}) : G_U(\mathbf{X}, \mathbf{Y}) \leq 0, G_L(\mathbf{X}, \mathbf{Y}) \leq 0, \mathbf{X} \in R^s, \mathbf{Y} \in R^t\}$. The projection of Ω onto the upper-level design space gives the feasible set for **X**, i.e., $U = \{\mathbf{X} \in R^s : \exists \mathbf{Y} \in R^t, (\mathbf{X}, \mathbf{Y}) \in \Omega\}$.Then the lower-level rational reaction set for $\mathbf{X} \in U$ can be defined as $R(\mathbf{X}) = \{\mathbf{Y} \in R^t : \mathbf{Y} \in \arg\min\{F_L(\mathbf{X}, \bar{\mathbf{Y}}) : G_L(\mathbf{X}, \bar{\mathbf{Y}}) \leq 0\}\}$. A feasible set that at least when the lover-level optimization model has a unique optimal solution for all values of **X** is called an inducible region, which is defined as:

$$IR = \{(\mathbf{X}, \mathbf{Y}) : (\mathbf{X}, \mathbf{Y}) \in \Omega, \mathbf{Y} \in R(\mathbf{X})\} . \tag{4.2}$$

Additionally, with the assumption that $R(\mathbf{X})$ is single-valued, which implies that there exists a unique response function $\mathbf{Y} = \mathbf{Y}'(\mathbf{X})$, an optimal solution to Eq. (4.1) can be found, which is denoted as $(\mathbf{X}^*, \mathbf{Y}^*)$.

In general, the solution of the game theoretic optimization leader-follower model can be realized in three steps:

Step 1: The leader makes a decision $\mathbf{X}'$ according to the leader's own strategy $F_U(\mathbf{X}, \mathbf{Y})$, then announces the decision to the follower with a set of constraints;

Step 2: The follower makes a decision subject to his own strategy F_L together with the leader's decision, then feedback the follower's feasible solution $\mathbf{Y} = \mathbf{Y}(\mathbf{X}')$ to the leader;

Step 3: The leader adjusts its decision to obtain a new $\mathbf{X}''$ base on the follower's feasible solution.

These steps are iterated until a satisfactory result is arrived for both the leader and the follower (Ji et al., 2013). The game theoretic optimization problems coincide with two-stage stochastic programming with recourse (Shapiro, 2006).Whereas deterministic optimization problems are formulated with known parameters, real world problems almost invariably include some unknown parameters. In a deterministic optimization problem, the goal is to find a solution which is feasible for all such data and optimal in some sense. Stochastic programming models are similar in style but take advantage of the fact that probability distributions governing the data are known or can be estimated. The goal here is to find some policy that is feasible for all (or almost all) the possible data instances and maximizes the expectation of some function of the decisions and the random variables. More generally, such models are formulated, solved analytically or numerically, and analyzed in order to provide useful information to a decision-maker. In the decision process, two different kinds of decisions are distinguished. The first stage deals with "here-and-now" decisions. Hence, decision variables representing "here-and-now" decisions do not depend on each single realization of the random variables. The second-stage corresponds to "wait-and-see" decisions that are made after knowing the actual values of the random variables. Consequently, these decisions depend on each plausible realization of the random variables. If random variables are represented by a set of scenarios, a second-stage decision variable should be defined for each scenario considered. The above steps, wherein the decisions are made after uncertainty is cleared, thus constitute a recourse problem.

4.5 Game Theoretic Optimization Framework for High Variety Assembly System Design

In order to identify a practical game theoretic optimization framework, the original Stackelberg competition game model will be analyzed and used as a foundation. Then with the general assumptions of high variety assembly system design problems, the framework suitable for this research will be developed and applied to each stages of assembly system design decision making in the following chapters.

4.5.1 Stackelberg Competition Model

The Stackelberg competition model is a strategic game in economics in which the leader firm moves first and then the follower firms move sequentially, which can be solved to find Nash equilibrium. The Nash equilibrium is a solution concept of a non-cooperative game involving two or more players in which each player is assumed to know the equilibrium strategies of the other players, and no player has anything to gain by changing only his or her own strategy (Osborne and Rubinstein, 1994).

To demonstrate the Stackelberg competition model, we can construct a simplified market model. Assuming there are two companies producing a similar product which meets identical customer demand. One of the companies is the leader who will make its decision first, by determining its production quantity q_1. The other company is follower who will observe the leader's action and make decision based on the leader's output, their production quantity is denoted as q_2. By using a simple linear price model $P(Q) = a - bQ$, the product's market price can be rewrite as $P(q_1, q_2) = a - bq_1 - bq_2$. Assuming the two companies total cost can be presented by $C(q_1) = c_1 q_1$ and $C(q_2) = c_2 q_2$, and using the economic profit function $\pi = P(Q) \cdot Q - C(Q)$, the maximal profit of the two companies from selling the product would be:

$$Max_{q_1} \pi_1 = q_1(a - bq_1 - bq_2 - c_1); \tag{4.3}$$

$$Max_{q_2} \pi_2 = q_2(a - bq_1 - bq_2 - c_2). \tag{4.4}$$

In order to find the Nash equilibrium, it is typically to start at the end of the game and work backwards. Since the leader moves first, we can take q_1 as given, find the best response q_2 and then back-up and consider the leader's choice q_1. In this case, it is obvious that the optimal solution to the follower is:

$$q_2^* = \frac{a - q_1 - c_2}{2}. \tag{4.5}$$

Substitute Eq. (4.5) into Eq. (4.3), a solution to this problem can be obtained.

However, in the real-life cases of assembly system design, most of the design evaluation criteria are implicit and cannot solved analytically. Furthermore, many design solutions are combinational sets of assembly processes or workstations which don't have a mathematical equation form. In order to solve such problems, a more generic numerical solution framework should be used. To demonstrate the numerical solution to this problem, the two companies Stackelberg competition model example are further solved numerically, the solution can be found by following steps:

Step 1: Leader selects a value of follower's production quantity q_2^0 within the constraint;

Step 2: Leader solves the upper-level optimization problem $Max_{q_1}\pi_1 = q_1(a - bq_1 - bq_2^{n-1} - c_1)$ to find the optimal solution q_1^n, n equals to current iteration level;

Step 3: Follower uses leader's solution as input and solves the lower-level optimization problem $Max_{q_2}\pi_2 = q_2(a - bq_1^n - bq_2 - c_2)$ to find the optimal solution q_2^n, n equals to current iteration level;

Step 4: If $q_1^n - q_1^{n-1} \le \varepsilon$ and $q_2^n - q_2^{n-1} \le \varepsilon$, where ε is the convergence limit, the iteration will stop and q_1^n, q_2^n is the Nash equilibrium solution to this problem. Otherwise, go to Step 2.

By assuming $a = 5000$, $b = 1$ and using different c values, the results of the two company Stackelberg competition model solution are shown in Figure 4.2.

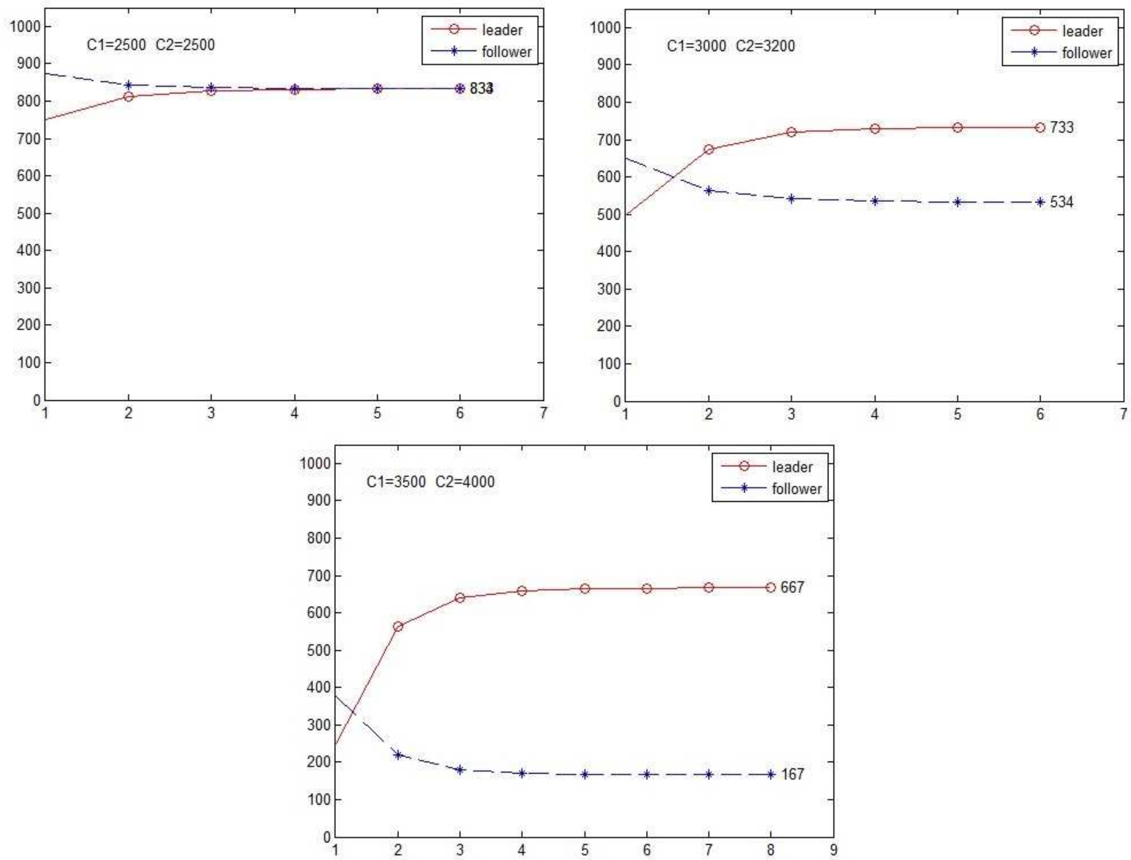


Figure 4.2 Numerical solution to a two company Stackelberg competition model

This example is a simplified game theoretic optimization problem, which actually would be faster to find an analytic solution than to find a numerical solution. However, when the cases move to the high variety assembly system design, most cases would be depending on the numerical solution framework as there is no analytic solution or its computational load is even higher.

4.5.2 Game Theoretic Optimization Framework

In conclusion of the solution process to a general game theoretic optimization problem, Figure 4.3 shows a proposed generic solution framework for game theoretic optimization.

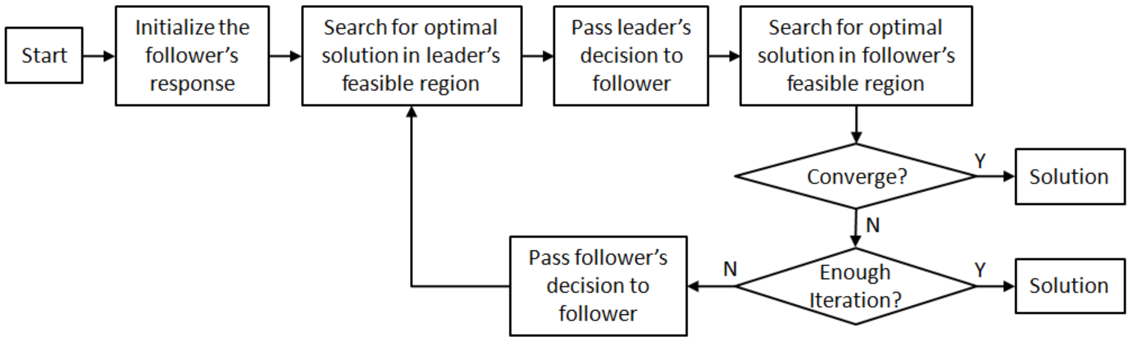


Figure 4.3 Game theoretic leader-follower optimization framework

The initial assumption of follower's response would affect the starting point of the optimization in entire feasible region, and thus may affect the computation time for reaching convergence and may also lead to different optimization result if there is local optima exists. The traditional way is to start from a zero vector or origin. However, using domain knowledge to interpret the follower's response and start from the "best guess" can help to reduce computation time in certain cases. Using a simplified evaluation of the objective function could also help to find a better starting point, although which is limited in most complicated cases.

In terms of searching for optimal solutions in both leader's and follower's optimization problem, there are many algorithm that could be used. In design and decision making of high variety assembly systems, the most common design variables are process sequences, workstation sequences, job orders and so on, which is mainly combinational variables. In this kind of optimization problems, GA is known to consistently produce better results compared to those produced by other techniques. In Chapter 5, the use of GA to solve game theoretic optimization problems for assembly systems will be further discussed.

The search algorithm will run multiple iterations until a stopping criteria is met. The stopping criteria could either be a maximum iteration number or a convergence is reached. Due to the implicit nature of optimizations on assembly system design, traditional convergence detection methods such as gradient norm condition or KKT condition is not applicable. A convergence limit can be used to determine if the search algorithm reaches a convergence with certain tolerance. The convergence limit will need to be set based on optimization parameters of GA, as different mutation rate and crossover rate will affect convergence speed. A bad convergence limit could cause the search algorithm to be stopped too early or never reach convergence.

4.6 Summary

The original achievements in this chapter can be concluded as followings:

1) The leader-follower decision model and mathematical model is suitable to describe the non-cooperative game in high variety assembly system design;
2) The numerical solution method proposed is capable to solve the implicit leader-follower optimization problem using Matlab.

As a summary, this chapter proposed a generic game theoretic optimization framework to solve high variety assembly system design decision making problems. Based on the nature of assembly system design, numerical solution with search algorithm is selected to solve the combinational decision making problem. An example of solution to a traditional game theory problem is used to show the basic concept, further applications and cases of this optimization framework will be discussed in the next several chapters.

CHAPTER 5

SOLUTION TO GAME THEORTIC OPTIMIZATION

To find an optimal solution of the game theoretic optimization based on the numerical solution framework introduced in last chapter, a suitable search algorithm is needed. Search algorithms define a design optimization problem in terms of a search problem where the search space is a space filled with a set of points. Each point in that space defines a solution, which will be either an assembly process sequence or a machine utilization plan in this research. The design optimization problem is then transformed into the problem of searching for the best solutions somewhere in the space of valid ones. The whole procedure is composed of three steps, which are: (1) define the problem and search space, (2) fix the goal of search and (3) use a method to reach this goal.

5.1 Genetic Algorithm Formulation

There are several reasons for using GA for design problems. GA is just one of many methods known in computer science (Pirlot, 1992). It is not easy to define exactly which of these methods is best for the game theoretic optimization problems. However, it is possible to identify methods that can consistently produce better results compared to those produced by other techniques. Rather than spending time, and effort developing new specialized techniques for new problems, most developers prefer to reuse proven algorithms.

GA is a stochastic global search optimization algorithm that mimics Darwin's theory of biological evolution. The idea behind GA is to use this power of evolution to solve optimization problems. GA works on the composition of genetic traits called chromosomes, in which successive operations through crossover or mutation give rise to better performing off-springs due to successive refinement of these hereditary traits. GA works with a population of design solutions and tries to find the best solution. Search algorithms like SA, PSO, ACO are also used for optimization but GA is most popular as it is more efficient and consistent in solving combinational problems which is common in assembly system related design. GA is also used for solving discrete optimization problems with application in statistics for the variable selection problems in regression and other multivariate statistical methods. It has been widely employed for tackling problems related to manufacturing network design, logistics and shop-floor scheduling problems.

5.1.1 Representation of GA

In tackling a search problem over certain space of possible solutions, it is necessary to construct a representation of the possible solutions for manipulation and storage. Thus, before applying a GA to any design problem, a certain mapping between the design combinations and the evolutionary method solution points must be made. In order to facilitate the use of GA, some terminologies must be introduced. The points in the search space are known as phenotypes, while their representatives in the solution space are known as genotypes. The structures used to represent genotypes are known variously as genomes or chromosomes. The genotype specifically refers to an individual’s genetic structure. The phenotype refers to the observable appearance of an individual. The process of producing a phenotype from a genotype is known as morphogenesis (Rekiek and Delchambre,2006). As shown in Figure 5.1, the mapping between genotypes and phenotypes reflects the mapping between physical assembly solutions and GA chromosomes combinations in this research.

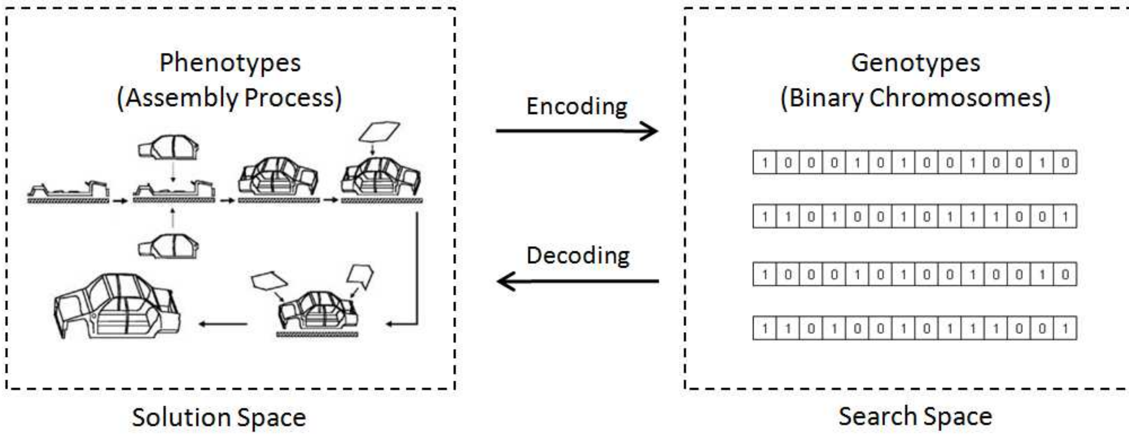


Figure 5.1 Mapping between solution space and search space

Generally, the phenotype will be encoded into a finite-length string called chromosome. And each element of the string which is called a gene will represent a construction unit of the phenotype. In order to achieve an efficient use of GAs, the encoding must be adapted to the particular search problem at first. Using a good representation is the first step to narrow the gap between theory and practice in the context of engineering optimization (Culberson, 1998). In the case of high variety assembly system design, it would be a single assembly process or a work station.

Obviously, the search result from the mapped search space would have chances to be infeasible in the solution space. To deal with this problem, there are four different basic strategies: rejection, repair, modifying the genetic operator, and assigning penalties (Gen and Cheng, 2000).The rejection strategy simply discards all infeasible individuals, which will apply to all GAs, but leads to a worse efficiency and sometimes leads to high computational complexity. The repairing strategy attempts to create only feasible solutions. For some problems, genetic operators can be modified so that they create only feasible solutions. Finally, penalty functions can be used when infeasible solutions can be recombined to form feasible ones. As GA is manipulating a coding of the solutions, not solutions themselves, it is obvious that the one-to-one correspondence mapping is the best one in which each solution is represented by exactly one chromosome and each chromosome decodes exactly one solution of the original problem. The n-to-1 mapping suffers from a lack of detail because some information is hidden from the GA.

5.1.2 General Solution Processes

All the variations of standard GAs are united by a common thread. The GAs work with a certain number of chromosomes. The set of individuals (solutions, chromosomes) of each generation is called a population. Chromosomes are characterized by their fitness and evolve through successive iterations (generations). A population of solutions is maintained and the evolution plays the role of adaptation of a population to its environment. This adaptation causes the creation of individuals of increasingly higher 'fitness'. The best solutions are selected for reproduction of every generation and the offspring are then generated from these fit parents using crossover and mutation. Thus, evolution drives the population of better individuals (John, 1992).

The standard GA solution process can be summarized by the following steps:

Step 1: Construct the GA encoding to represent the solution space. GA can operate on any data type (representation) which determines the bounds of the search space. It is desirable that the representation can only encode feasible solutions, so that the objective function (fitness) measures only optimality and not feasibility;

Step 2: The initial population is created during an initialization phase and it is often generated randomly. Generally, some domain knowledge is used by GA to start the search from a promising region of the search space;

Step 3: Every member of the population is then evaluated and a fitness value is given according to how well it fulfills the objectives. A fitness function is used to perform the evaluation in order to find better solutions;

Step 4: The GA selects individuals with a higher overall fitness when picking ‘parents’ from the population. Then, these fitness scores are used to determine which individuals will have more probability to participate in creating the new population;

Step 5: Based on the fitness values, the GA selects candidate solutions and combines (crossover) the best traits of the parents to produce superior children;

Step 6: A small part of the population is mutated. Single existing individuals are modified to produce a single new one. However, sometimes it is likely to produce harmful or even destructive changes than beneficial ones;

Step 7: Natural selection ensures that the weakest creatures die, or at least do not reproduce as successfully as the stronger ones. In the same way, a population is maintained with the fittest solutions being selected for reproduction. New generations are formed by selecting some parents and offspring and rejecting the less-fit ones;

Step 8: A generation is a population at a particular iteration of the loop. This iterative process (selection, crossover, etc.) continues until the specified number of generations is passed, or an acceptable solution has emerged.

5.2 Leader-follower GA for Game Theoretic Optimization

As the GA is more capable of solving the multi-model optimization problems which are always seen in the real life, comparing with traditional calculus-based or approximation-based optimization techniques, GA excel in solving combinatorial optimization problems. Taking the advantage of GA with necessary modifications, I can develop a solution schema for game theoretic optimization of high variety assembly system design, namely the Leader-follower Genetic Algorithm (LFGA).

5.2.1 Quantification and Encoding

In order to be consistent with the game theoretic leader-follower decision making, two kinds of chromosomes are composed: the Upper-level Chromosome (ULC) and the Lower-level Chromosome (LLC). Each chromosome is a representation of a solution to the optimization problem. In a game theoretic optimization, a selected ULC would normally leads to several LLCs and follower should select the best LLC as response to the leader. The encoding scheme is shown in Figure 5.2.

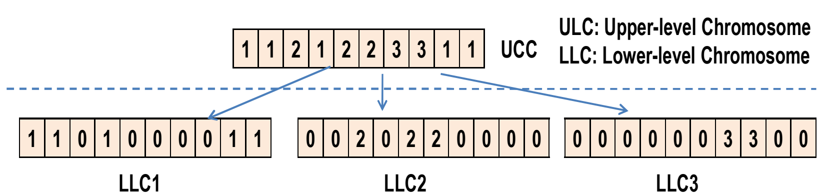


Figure 5.2 LFGA encoding scheme

Take the high variety process design and process planning optimization problem as an example. Given a product family that has a number of *m* component parts to be assembled, each product variant could have multiple assembly process designs. First, the process designer creates multiple assembly processes for each variant, forming a process set. When all part process sets are generated, he could randomly pick a process plan in each set to generate an assembly process solution. Then he should evaluate all process solutions and sends the one producing the highest process commonality index to the process planner for further resource allocation and part sequencing. Among all resource allocation plans and part processing sequences, the process planner could find a solution which can achieve a maximal resource utilization rate. But this solution will influence setup time and sequencing flexibility, and in turn affect the value of process commonality index that is relevant to the process designer’s concern. The process designer and the process planner constitute a typical leader-follower decision-making scenario, whereby the leader’s goal is to maximize process commonality and the follower’s goal is to maximize resource utilization. Iteration between the leader and the follower coincide with the game theoretic optimization of process commonality and resource utilization.

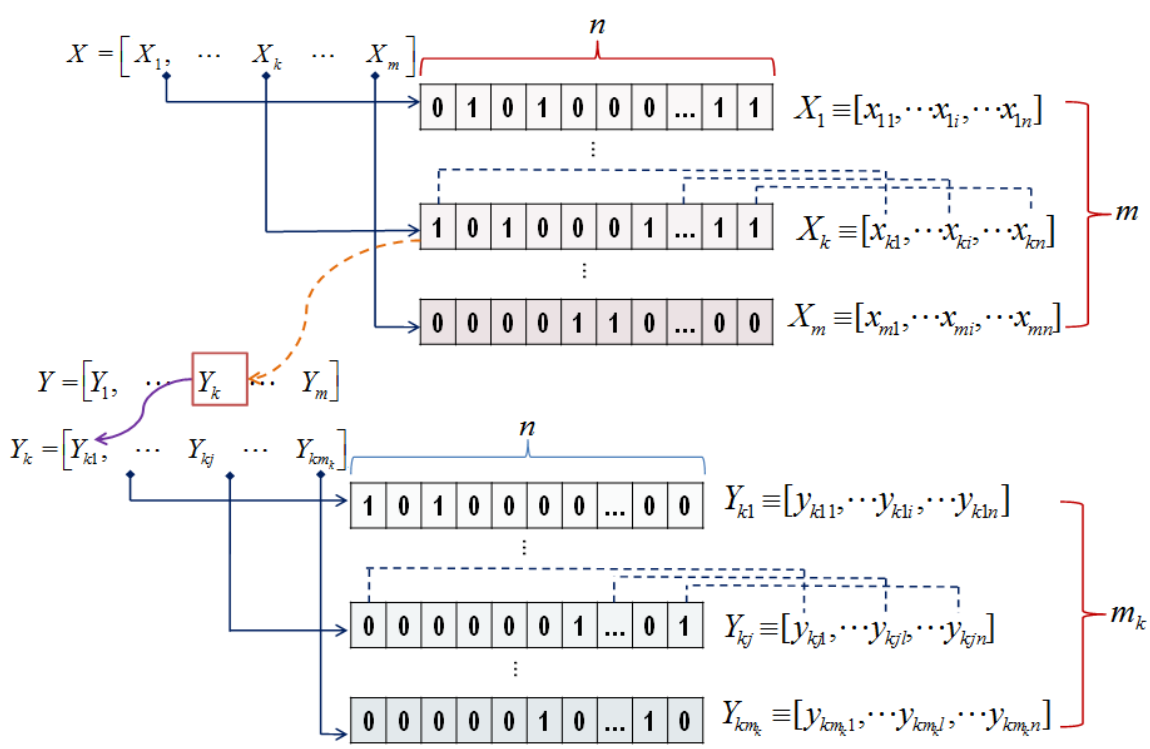


Figure 5.3 LFGA encoding for process design and planning game theoretic optimization

As shown in Figure 5.3, the ULC is a selected process solution for the entire product family, which is denoted as $\mathbf{X}=\{\mathbf{X}_1,\cdots\mathbf{X}_k,\cdots\mathbf{X}_m\}$, where $\mathbf{X}_k$ is the assembly process design of the *k*-th product variant. One or several elements in $\mathbf{X}_k$ represents a certain assembly process to assemble the *k*-th product variant. Normally each process design will results in one or several assembly process plans and resource allocation plans. A selected process plan and resource allocation plan shown in LLC is denoted as $\mathbf{Y}_k=\{\mathbf{Y}_1,\cdots\mathbf{Y}_k,\cdots\mathbf{Y}_m\}$, where $\mathbf{Y}_k$ is the selected process plan for the *k*-th product variant. Assuming there will be m_k work stations involved the assembly of product variant *k*, $\mathbf{Y}_{kj}\,(j\in[1,m_k])$ represents the resource allocation plan of work station *j* for product variant *k*.

The encoding of genes which form the fundamental elements of ULC and LLC are determined on the domain problems. In the case of assembly process design, it will be the assembly process ID number in binary format. And in the case of assembly process planning, it will be work station ID or machine ID in binary format.

5.2.2 Initialization and Population

At the beginning, the population is generated randomly. Corresponding to the game theoretic optimization framework, it involves two initialization stages, i.e.,the upper- and lower-level initializations. At each level, after the initialization, the evolution should make extremely rapid progress at first. Indeed, most solutions are largely different and belong to different areas of the search space. Over time, the population begins to converge, with the separate individuals resembling each other more and more. The LFGA narrows its search in the solution space and reduces the changes made by evolution until eventually the population converges to a single solution.

An important parameter in this step is the population size. Normally the population size is kept as a fixed number so the diversity is guaranteed in a certain level. A number of population updating modes are used in GA. The main approaches are the steady-state update and generational update. A generational update scheme is a population maintenance mechanism in which N children are produced from a population with size N to form the population at the next time-step. This new population of children completely replaces the parent population. In contrast, in the steady-state approach, a single child is produced at each time-step which replaces a single member of the old population. The most straightforward way to maintain population diversity is to increase the population size. In large size problems, however, restrictions on the computer resources, such as time and memory, make it infeasible to run GA with the population size needed to maintain the required diversity. In the cases of the following chapters, the generational update method will be used as it has a much faster convergence speed, which is very important for game theoretic optimization solutions, as the leader-follower decision structure based LFGA have more iterations than normal GA.

5.2.3 Fitness Function and Selection Method

A fitness function is necessary to evaluate the fitness value of each chromosome within the population of each generation. Good chromosomes should be exposed to more opportunities to be selected as a feasible solution, whereas poor ones should not be selected at all. Generally speaking, the fitness function is the evaluation criteria that determines which solution should be selected to have more chance to be kept until the end of GA, when only the optimal solution is left. In an optimization problem, such criteria will be the goal function. A fitness landscape is a set of points in n-dimension space (hyper-surface) obtained by applying the fitness function to every point in the search space. To optimize a function efficiently, the fitness function must be clearly defined and higher fitness individuals must be explicitly promoted. Sometimes a certain simplification is needed as this fitness function will be used a lot in the iterations of LFGA. As a result, if any other ability besides high fitness is desired, LFGA must directly encourage the formation of individuals with the desired ability.

Darwin defined natural selection or survival of the fittest as ‘the preservation of favorable individual differences and variations, and the destruction of those that are injurious...’ (Darwin, 1963). When selection is the only mechanism in GA between generations, the best individual is eventually selected to completely take over the population. The selection mechanism determines which individuals will have all or some of their genetic material passed to the next generation. There are many selection methods in GA, such as the elitist model (DeJong, 1975), which tries to reduce the stochastic errors of the selection mechanism. This is done by introducing a count for each solution *s*, initially set to the $f(s)/\overline{f}$ value ($f(s)$ is the fitness value of solution s and $\overline{f}$ is the average fitness of the population) and decreased by 0.5 or 1 each time the solution is selected for reproduction with crossover or mutation respectively. Thus, when a chromosome count falls below zero, the solution is no longer available for selection. The more widely used selection method in GA is the Roulette Wheel Selection (RWS) method (Golberg, 1989). The RWS technique works like a roulette wheel in which each slot on the wheel represents an individual of the population. The size of each slot is proportional to the corresponding individual fitness, which implies the probability of being selected is proportional to the quality of the solution. Another method is called ranking selection (Fonseca, 1995), which is similar to the RWS, however, the solutions are selected proportionally to their rank rather than to their evaluation. As shown in Figure 5.4, for five different individuals, RWS selection probability is based on their fitness, and ranking selection probability is based on their rank and ranking distribution.

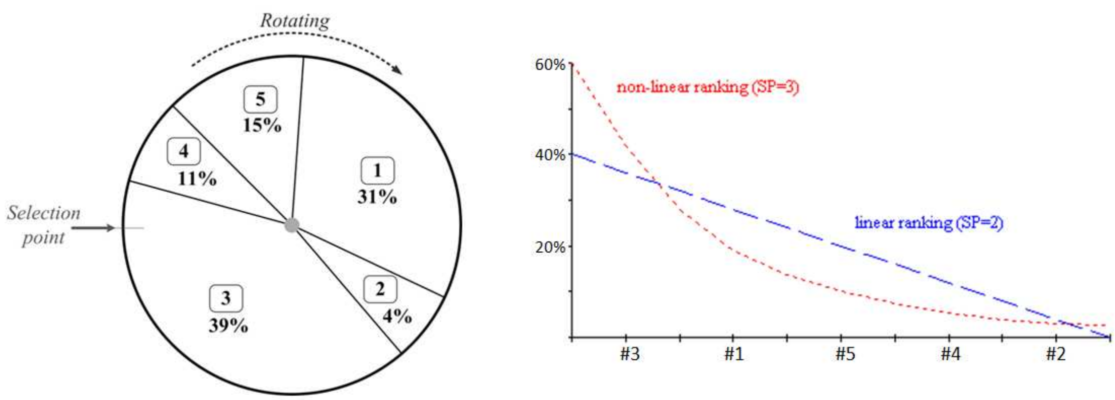


Figure 5.4 Comparison of RWS and ranking selection method

The trade-off between exploitation and exploration is generally viewed as one of the key features in an effective search. It is widely accepted that a higher selection pressure leads to fast convergence, but also increases the likelihood of premature convergence, which leads to a local optima. On the other hand, very low selection pressure increases the run-time and can causes the failure to improve solutions. Among the mentioned methods, the elitist model has the highest convergence speed but the threshold factor is hard to leverage in applications. The ranking selection method will help to explore more feasible regions but turns to be hard to converge. This is because the lower fitness population will always have certain chances to be selected no matter how low their evaluation values are. RWS is used in this research as it shows the best leverage amount all the selection methods.

5.2.4 Genetic Operators

The selection mechanism does not introduce any new solutions for consideration from the search space. It just copies some solutions to form an intermediate population. The second step of the evolution cycle is the recombination which takes the responsibility of introducing new individuals into the population. This is done by the genetic operators: crossover, mutation and inversion. Thus, together, crossover, mutation and inversion allow LFGA to discover fit, short and low-order solutions over time.

The most popular mechanism is where two individuals are selected and are crossed over in order to produce new offspring at each iteration. The aim of crossover is to produce new solutions in regions of search space where successful ones have already been found. There are many variations of the crossover operators, the most common ones are the ‘P-point crossover’ and the ‘uniform crossover’ (Rekiek and Delchambre, 2006). In the P-point crossover, each parent is divided at P locations into P+1 contiguous sections, numbered from 1 to P+1. Two offspring are created by exchanging every odd section between the two parents. The uniform crossover can be thought of as a P-point crossover, where P+1 is the number of genes in each parent. Therefore, each gene is a section and every section is probabilistically interchanged between the two parents. The crossover gives LFGA an advantage to perform better than other algorithms. Without crossover, LFGA lacks the additional instruments in the SA or the Tabu Search (TS) like temperature, Tabu list.

Mutation is a mechanism that has only a small chance of occurring in LFGA, which tends to produce more infeasible results than with normal GA due to the combinational nature in LFGA encoding. The standard mutation operator randomly perturbs offspring composition by changing a small number of alleles. Unlike crossover, mutation is a unary operator, and only acts on a single individual at one time. Some GA uses only the mutation operator and do not perform any recombination. These GA are roughly equivalent to running many SA algorithms in parallel. Mutation maintains diversity in the population and thus helps LFGA to reduce the chance resulting local optima.

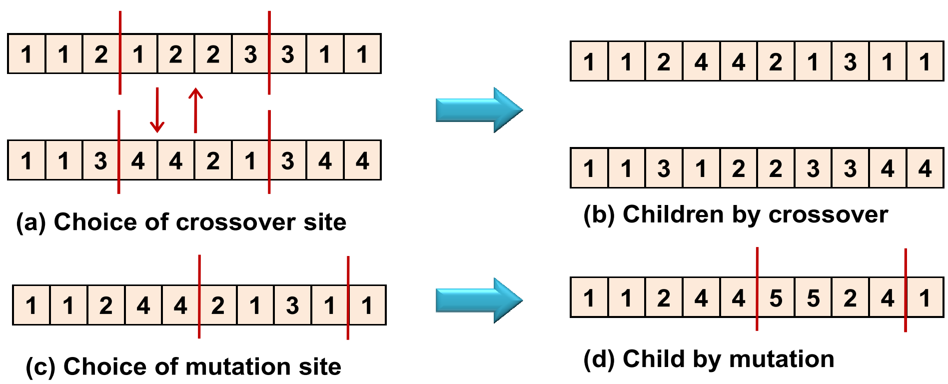


Figure 5.5 Crossover and mutation operation

Inversion is used to mitigate a drawback of the crossover operator. Since the crossing sites are picked at random, longer solution sets are disrupted more often than shorter ones. Whenever one of the crossing sites falls between the genes which define a solution set, the child will inherit only a part of the solution set. This has a very high chance to happen in LFGA as many of the solution sets in assembly systems are long gene sets and an infeasible solution is very likely to be reached by cutting these sets. Inversion allows shortening of long solution set by rearranging the positions of loci on the chromosome. In the standard inversion operator, two sites are selected at random and the order of the loci between the sites is reversed.

5.2.5 GA Solution Process for Game Theoretic Optimization

The solution process of LFGA for game theoretic optimization is shown in Figure 5.6. The initial population of both ULC and LLC will be done at the same step, this guaranties the upper-level fitness function to have a LLC input in the first iteration. Although a random initial population on both ULC and LLC may lead to some infeasible solutions, both ULC and LLC fitness function would have penalties with consideration of infeasible solutions to make sure the number of selected infeasible individual is kept to minimal.

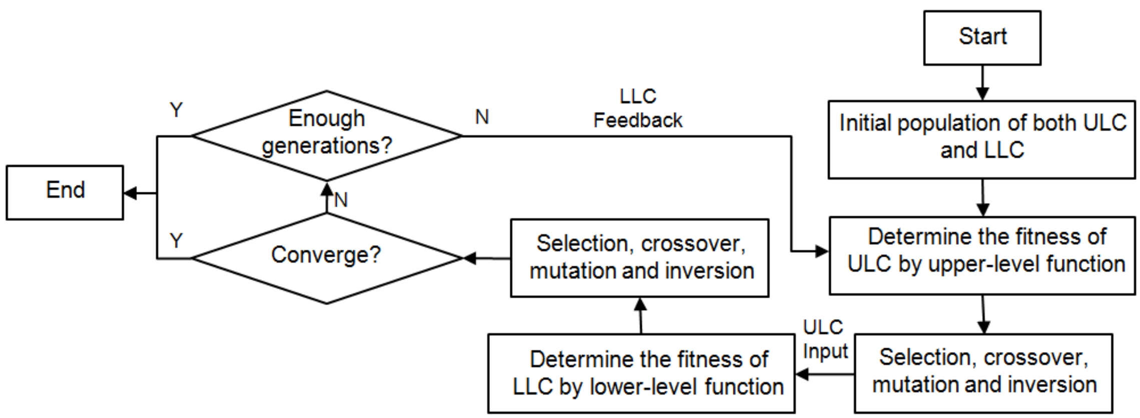


Figure 5.6 LFGA solution process flow chart

After the fitness of all the ULC solutions is determined, a RWS based selection will be used to create the next generation of ULC. Crossover will also use the same fitness scores to produce certain numbers of new individuals. A small portion of the new ULC population would mutate and inverse, the parameter of crossover, mutation and inversion would be determined by a sensitivity study of the domain problem. And further tweak of these parameters may also happen if the solution converges too fast or too slow.

After that, the new generation of ULC will be passed to the lower-level for LLC fitness calculation, and then followed by RWS selection, crossover, mutation and inversion, which is similar to ULC. When the new generation of LLC is produced, both ULC and LLC will be compared with the last several generations of them for convergence test. The whole GA search iteration will continue as long as the convergence condition is not reached, and the number of generation is within the limit.

5.3 Discussions

The domain problems in assembly system design is a good example of combinational problems, which deals with decision making among different combination of assembly methods, sequences, manufacturing resources and material flows, etc. Comparing with traditional calculus-based or approximation-based optimization techniques, GA excels in solving combinatorial optimization problems. Traditional GA is designed to meet an ultimate goal with an overall governing fitness function, which works well in all-in-one optimization problems. However, such all-in-one solutions assume that assembly process design and process planning are homogenous problems, which can hardly capture the inherent difference underlying two heterogeneous decision-making problems, whereby different sets of objectives are involved and often conflicting in problem solving per se (Jiao and Tseng, 2013). The game theoretic optimization model is developed to identify this equilibrium solution, as opposite to the traditional all-in-one multi-objective optimal solutions that deals with an ideal case and assumes unlimited production capability. Then the modification of traditional GA becomes a necessity to provide a practical solution method for game theoretic optimization. With the inherent leader-follower decision making schema applied to GA, the LFGA allows the conflicting nature between different parties in game theoretic optimization to be reflected in the evolution iterations pass between ULC and LLC. Considering the computational and organizational complexities in high variety assembly system design, LFGA uses a dual stage evolutional operation to make sure only one fitness evaluation is needed for all offspring at each stage. The trade-off between convergence speed and exploration range in LFGA is handled by the combination of RWS selection based on ULC/LLC fitness, relatively high crossover rate and low mutation rate.

5.4 Summary

The original achievements in this chapter can be concluded as followings:

1) By using a leader-follower hierarchical encoding, the modified GA is capable of representing the proposed game theoretic optimization problem;
2) Standard GA crossover operator is replaced by inversion operator, which keeps the long assembly process gene codes feasible;
3) The modified LFGA in Matlab presents the ability to solve game theoretic optimization problem of high variety assembly process design.

As a summary, the proposed LFGA provides a practical solution to game theoretic optimization problems which is identified in Chapter 4. And the technical challenges brought by high variety assembly design would also come to a feasible quantitative solution using the leader-follower decision making model and LFGA method. Following the framework brought by game theoretic optimization model and FLGA solution, the high variety assembly design problem will be analyzed in the following chapters.

CHAPTER 6

ASSEMBLY PROCESS VARIETY DERIVATION

6.1 Introduction

As the backdrop of product families, a well-planned architecture (the conceptual structure and overall logical organization of generating a family of products) will provide a generic umbrella to capture and utilize commonality, within which each new product instantiated and extends so as to anchor future designs to a common product line structure (Du et al., 2001). A number of perspectives on product platform and representation exist in literature. A review suggests that product platform has been defined diversely, ranging from being general and abstract to being industry and product specific (Robertson and Ulrich, 1998).

There are two streams of research prevailing in the field of developing product platforms and representation (Meyer and Lehnerd, 1997). One perspective refers to development of a product platform as a physical one, namely a collection of elements shared by several related products. Accordingly, the major concern is how to identify common denominators for a range of products. This effort is geared towards the extraction of those common product elements, features, and/or sub-systems that are stable and well understood, so as to provide a basis for introducing value-added differentiating features and thus brings the possibility to produce high variety products based on product families. The other dominating perspective is to exploit the shared logic and cohesive architecture underlying a product platform. Such researches lead to later on development of more generic representation of product platforms. One endeavor towards product platform representation development is to design product families in the way of stretching and/or scaling.

Product data can be represented by a BOM that is used for an end product to state raw materials and intermediate parts or subassemblies required for making the product. On the other hand, production information is concerned with how a product is produced, that is, the specification of operations sequences to be performed at corresponding work centers along with related resources such as machines, labors, tools, fixtures and setups. Similar to describing a product structure using a BOM, an operations routing can be constructed to represent the production structure for a given product (Olsen et al., 1997). A product platform, consisting of diverse product variants, is characterized by a Generic Product Structure (GPdS) (Du et al., 2001). It is proposed to characterize the source of variety based on the hierarchical decomposition of product structures. Product variants can share a common structure, which may be common product technologies, modules or configuration mechanisms. GPdS acts as a generic data structure for such variants. Accordingly, its related production processes can be collated as standard routings in the form of a Generic Process Structure (GPcS). These standard routings form the basis of various process variations in consequence of product variety. As being identified in Chapter 3.2, the proposed methodology to solve the research problem includes:

1) Use generic representation which utilize the object-oriented data structure to represent the product and process variety, in order to enhance data structure efficiency, which is very critical for high variety assembly system;

2) Use data mining technique to identify the relationship between each product variety and assembly process variety, then generalize such relationships into a variety mapping for process variety propagation.

6.2 Generic Product Structure

Traditional BOM is a directed acyclic graph representing the composition of a product. Every node is an aggregation of its children nodes. In this graph, all end nodes are individual components, all intermediate nodes are partial assemblies and the root node is the final product. The graph arcs show the quantities of child components required to create a single instance of the parent. As shown in Figure 6.1, $\{X_1, X_2, X_3, X_4\}$ are individual components, $\{Y_1, Y_2\}$ are subassemblies and $\{Y_3\}$ is the final assembly. In terms of quantities, three units of $\{X_2\}$ are needed to assemble one unit of $\{Y_1\}$, two units of $\{Y_1\}$ are needed to assembly one unit of its parent subassembly $\{Y_2\}$, and two units of $\{Y_1\}$ are needed to assembly one unit of the final product $\{Y_3\}$. With the comparison of a tree graph of the same product in Figure 6.2, it is oblivious that BOM structure is more efficient in representing products with same components repeatedly used.

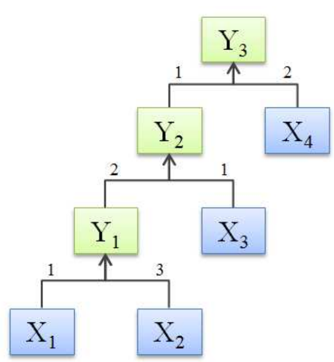


Figure 6.1 A traditional BOM structure

The BOM product structure has been widely used in industry as a standard product structure for decades. In dealing with variety, the traditional approach is to treat every variant as a separate product by specifying a unique BOM for each variant. This works with a low number of variants but not when customers are granted a high degree of freedom for specifying products. The problem is that a large number of BOM structures will occur in mass customization production, in which a wide range of combinations of product features may result in millions of variants for a single product. Design and maintenance of such a large number of complex data structures are difficult, if not impossible. To deal with a large number of variants, it is necessary to understand the implication of variety and to characterize variety effectively.

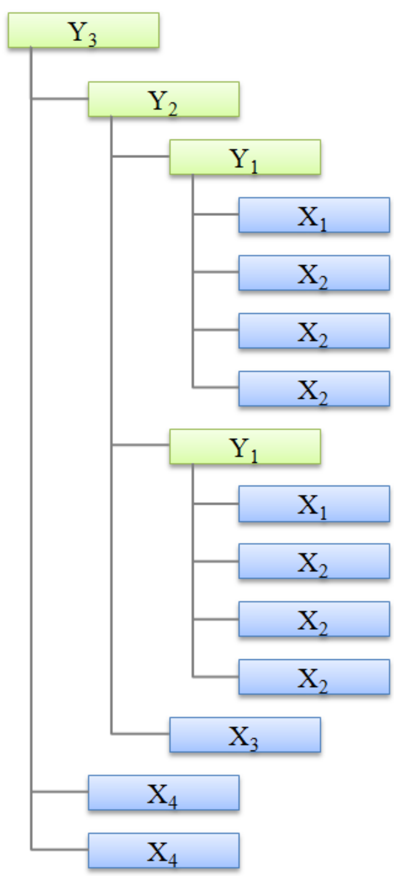


Figure 6.2 A product representation using product tree

Introducing a generic product structure might not reduce complexity at first but will give control over complexity. When product family with high variety is involved, the benefits of the generic product definition will become obvious. A generic product structure is of great value for product management when planning new product development, for research and development as input on what is needed to be new design and which design can be re-used, for production planning and not the least purchasing organization for procurement planning. All these disciplines produce information to a common structure and consume information from the same along the products lifecycle.

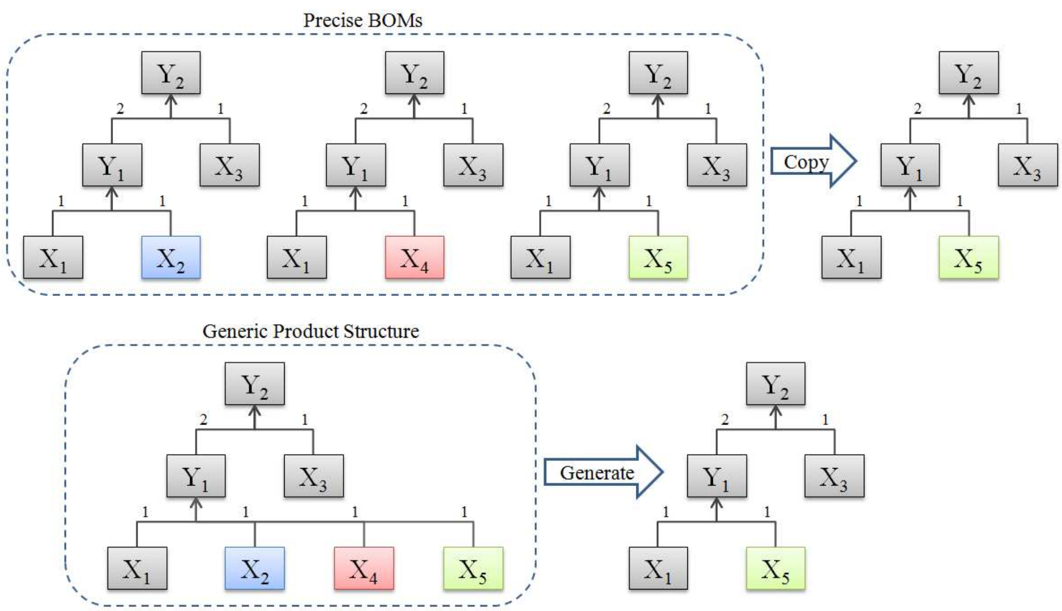


Figure 6.3 Comparison of BOM and GPdS for high variety products

GPdS is a hierarchy consisting of constituent items at different levels of abstraction, where items can be either abstract or physical entities. The physical entities in a GPdS, which are also named as modules in general, can be raw material, $\{R_i\}$, purchased part, $\{C_i\}$, intermediate part, $\{I_i\}$, and subassembly, $\{SA_i\}$. Some of them are primitive, which means they cannot be further decomposed, thus becoming a leaf node of the decomposition structure. A compound module is composed of primitive modules and/or other compound modules. Each module can possess several variants (instances of the same module type). The nesting of basic constructs is achieved by introducing compound module(s) as the component(s) of another compound module. In this sense, a nested GPdS can be regarded as a multi-level decomposition structure of compound modules.

The parent-child relationship between a parent module and child module is called a structural relationship $\{SR_i\}$. With respect to product structures, it is equivalent to the goes-into relationship defined for BOM structures (Van Veen and Wortmann, 1992). The structural relationship variants can only be either exist $(SR_i = 1)$ or not $(SR_i = 0)$. The existence of SR_i means that the child module is included as the component of parent module. Otherwise, it is excluded. Different variety generation can be implemented through defining such *SR* variants.

All variants of modules in GPdS are controlled at leaf nodes. This is because the variety of a compound module can be achieved through its primitive modules. Therefore, the relationship between variants and the corresponding module can be observed as instantiation of the module according to certain conditions. Such variants and module relationship are represented using include conditions $\{IC_i\}$.

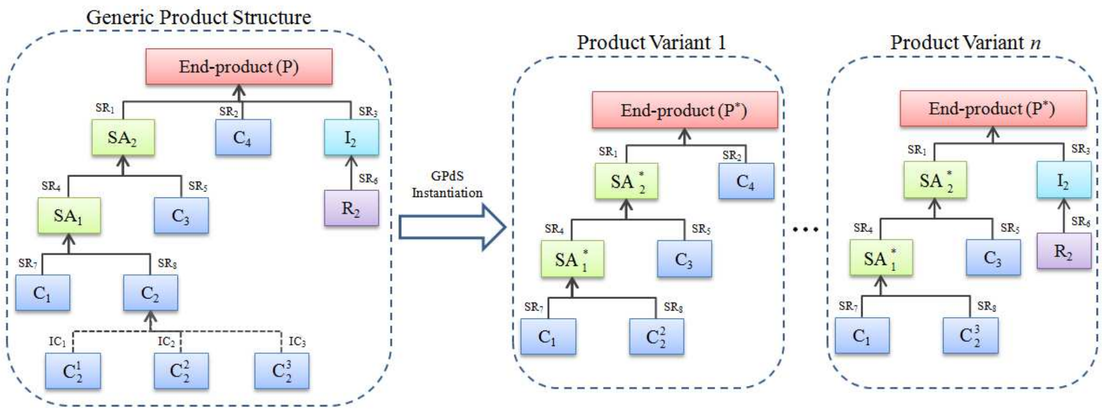


Figure 6.4 Variety derivation though GPdS instantiation

6.3 Generic Product and Process Structure

In practice, process information for an enterprise is often described in various forms of documents such as, product specifications, routing sheets, and job cards. These documents may be a suitable representation scheme for humans who must possess the knowledge to understand the information, but does not lend itself to formal analysis, monitoring, or improvement. Therefore, it is necessary to develop a modeling formalism. Such a formulation should provide a sufficiently powerful syntactic model to support rigorous analysis and manipulation of process variety, while facilitating the application of semantics to support process design enactment and detailed observations from a number of perspectives including customers, design and production (Mills and Tanik, 2000).

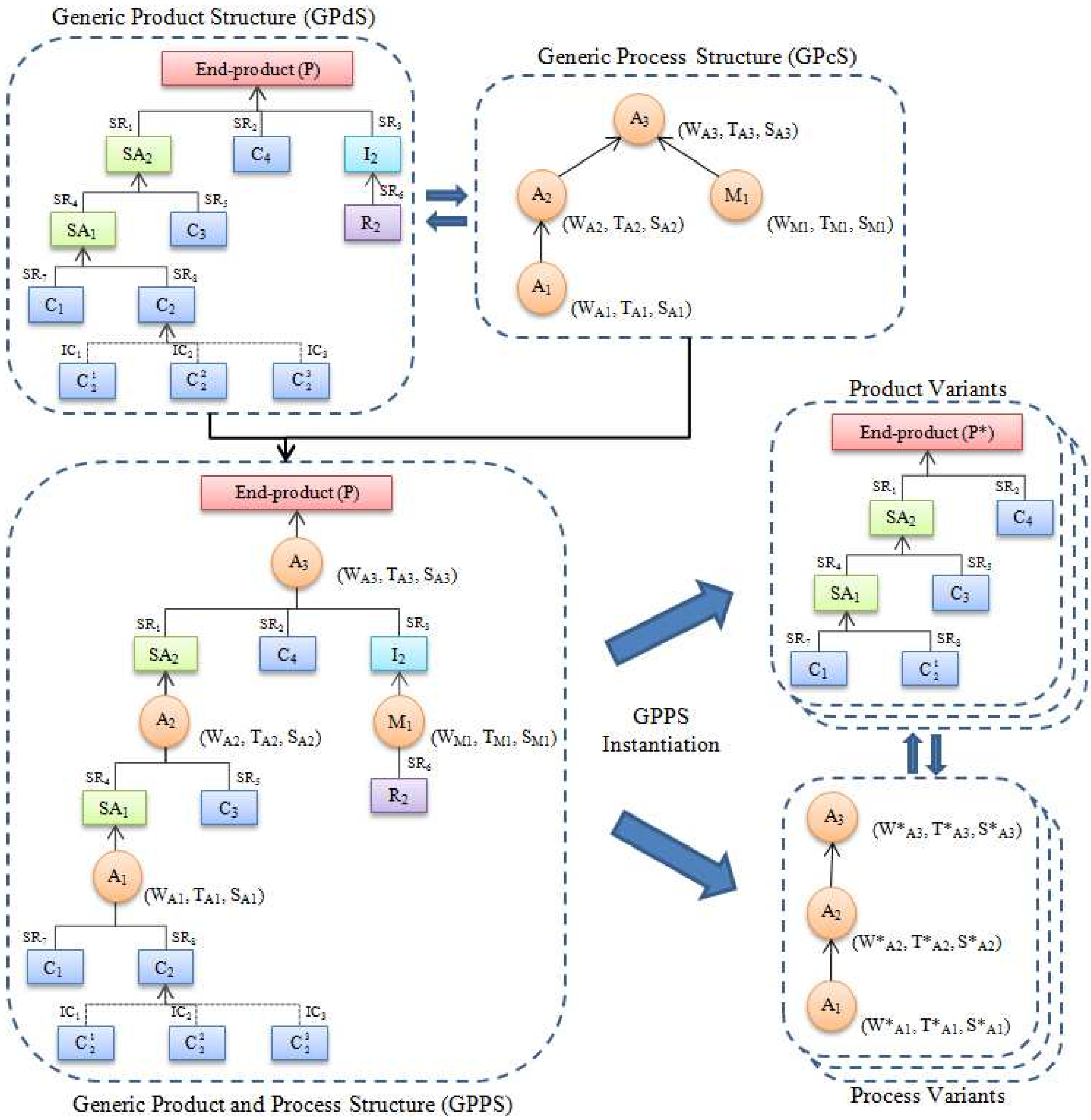


Figure 6.5 GPPS and its instantiation

Both GPdS and GPcS are rigorous syntactic models, and can be used to formulate product variety and process variety. In order to find the relationship between product variety and process variety, it is necessary to identify the relationship between the product structure and process structure, which are embodied in the materials required by particular production operations (Jiao et al., 2007). The link between product structure and process routing data can be established by specifying each component material in the product as required by the relevant operation of the routing for making its parent product (Mather, 1987). The material requirement and corresponding operation sequence links can synchronize GPdS and GPcS into a unified generic structure, which is called Generic Product and Process Structure (GPPS).

While the GPdS associates each component material directly with its parent product, a component material in the GPPS is associated with the relevant operation in the GPcS for producing its parent component. It reuses the elements of GPcS, including assembly operation $\{A_i\}$ and machining operation $\{M_i\}$. Each operation has operation parameters including work center number (*W*), cycle time (*T*) and setup number (*S*). For each manufactured end or intermediate product, a single-level GPPS can be derived by specifying the sequence of operations required for producing that product in connection with materials and resources (categorized in terms of work centers, cycle times, and setups) required for each operation. The multilevel GPPS can be composed by linking the single-level GPPSs of lower-level intermediate parts through the operations that require them. Taking advantage of the meta-structure inherent in the generic variety representation, variant derivation can be implemented through the instantiation of a GPPS with respect to the given values of particular variety parameters. For example, in Figure 6.5, the generic component C_2 has three variants. The generic identification of C_2 family is described as a set, $C_2=\{C_2^1, C_2^2, C_2^3\}$. The corresponding process variation of C_2 family involves a generic assembly operation A_1. Assuming different variants of C_2 family use same work center but different setup with different setup time, then the assembly operation A_1 can be described as $A_1^n(W_{A1}, T_{A1}^n, S_{A1}^n)$.

In a GPPS, GPdS and GPcS are unified not only by the material links, but also by their variety parameter sets and values to handle variety. Thus, the class-member relationships between generic items and their variant sets can be consistently used for both product and process variant derivation. In this way, the correspondence between product and process variety can be maintained throughout the variation of both product structures and routings.

6.4 Generic Product Variety Propagation

Variety propagation plays a very important role in GPPS. For products with high variety, it can be too expensive to define each one of these with a different product and process structure. Using the GPPS structure, it is possible to define all the product variations in a single GPPS structure simultaneously. In product families with high variety products, the syntactic GPPS model with variety parameters will help engineers to determine and analysis both product and process variety without repetitive works and as well as providing automation capabilities.

The generation of product variants in a product structure has been dealt with in some product family models, including such useful concepts as parameter inheritance (Wortmann and Erens, 1995; Jiao and Tseng, 1999), configuration constraints (Wortmann and Erens, 1995; Baldwin and Chung, 1995; Jiao and Tseng, 2000), and selection conditions (Wortmann and Erens, 1995; Baldwin and Chung, 1995). Nonetheless, the generative capability of these models is limited due to unclear handling of the underpinning structures of constraints across different views (Du et al., 2001). The variant derivation is summarized into following four steps in different product realization steps:

Step 1: Generate variant selection constraints during production specification stage based on customer requirements;

Step 2: Selected functional features are transformed to the variety parameters of the end product and then propagated down the hierarchy of the GPPS;

Step 3: Variants are propagated downward during the instantiation stage, suitable primitive variants are selected;

Step 4: Generate detailed product design based on the instantiated variant.

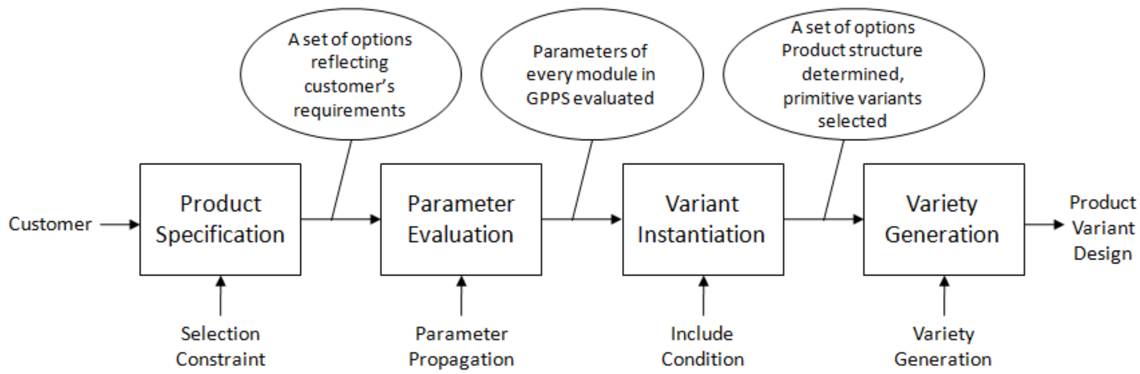


Figure 6.6 An illustration of the variant derivation process (Du et al., 2001)

Usually there is a set of attributes which are associated with each module, and some of them are variety parameters. The foundation of generic variety representation originates from object-oriented modeling. Along with behavior and identity, the concept of state is essential to the definition of an object in object-oriented modeling. Usually, the state of an object encompasses all static properties of the object as well as the dynamic values of these properties. Every object class defined in a GPPS can be regarded as a generic item. If these generic items are assumed to be analogous to the objects in object-oriented modeling, a concept of variety state should also hold true in playing a similar role in indirect identification of individual variants from generic items. These variety parameters can be inherited by child / primitive modules from a parent / compound module. Sometimes, variety could also come from adding or removing certain product modules / components. This kind of variety should also be reflected on the variety parameters of parent nodes which have option of add or remove modules. In this way, all variants of primitive modules are in fact organized by different instantiations of variety parameters (Du et al., 2001).

Within the axiomatic design framework, functional variety and technical variety correspond to customer needs and functional requirements in the customer and functional domains, respectively. The instantiation of a product / process variety is accompanied by the propagation of variety parameters from higher-level to lower-level modules. At the top level, all variety parameters are transformed from functional features and options (so called functional variety) and conveyed to the end-product. The allocation of these variety parameters to lower-level modules involves the relationships of variety parameters and differentiation enablers, and thus requires domain knowledge about mappings from functional requirements to design parameters (Suh, 2001). Some research has been conducted towards such variety fulfillment issues. Functional variety (commercial view characteristics) is linked to component families according to direct correspondence (Wortmann and Erens, 1995). Each child node inherits parameters from its immediate parent node (Jiao and Tseng, 1999). In both work, inheritance is assumed to be direct correspondence. However, more complicated scenarios exist. Functional variety may be in a different form from technical variety defined in terms of engineering parameters. For example, "A watch with date indication" is a functional variety, whereas the corresponding engineering parameters may be a set of parameters related to design concepts, assembly procedures and calibration processes. In addition, certain variety parameters of a lower-level module may be derivatives or functions of its parent's parameters.

6.5 Mapping between Product and Process Variety

Production planning of product families is enacted between the product and process domains and encompasses diverse product items in design to various process elements in production. Consequently, variety information is expressed as different sets of context for product differentiation and process variation. Variables used to describe product and process variety are often poorly understood and are usually expressed in abstract, subjective or conceptual terms, leading to working on the basis of vague assumptions and implicit inference (Jiao et al., 2008). Data mining lends itself to gaining knowledge from historical data, i.e. to reveal previously unknown and potentially useful patterns of product family design and production (Agard and Kusiak 2004). Harding et al. (2006) reviewed the progressive applications and potential of data mining techniques for manufacturing and production processes. Shahbaz et al. (2006) demonstrated the strength of association mining in improving product design, manufacturing process data and information management. The mapping relationships are embodied in association rules that can be subsequently deployed to facilitate process planning of product families while leveraging upon existing production processes. As shown in Figure 6.7, the data elements and their relationships are presented with product and process variety. The mapping relationships between product items and process elements are influenced by particular variety parameter instances originating from product differentiation.

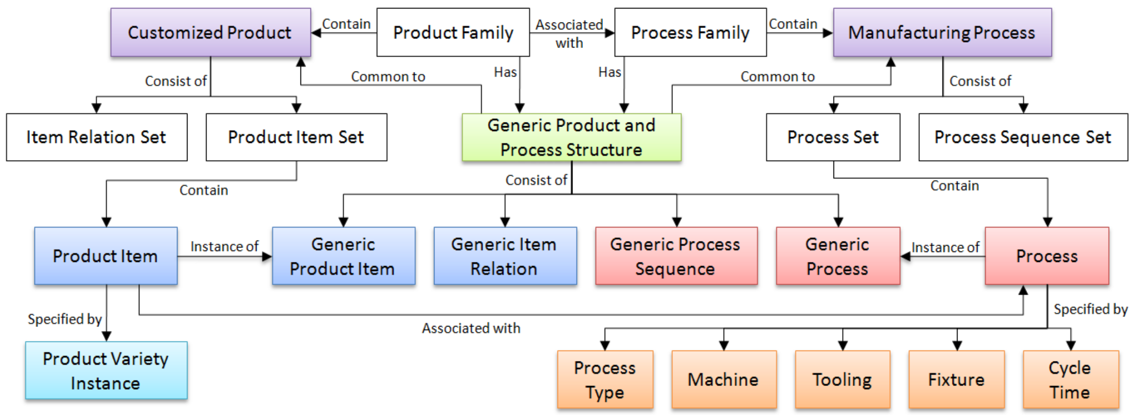


Figure 6.7 Entities and relationships associated with product and process variety

The mining of variety mapping rules starts with existing product and process data in the company's databases and information systems. Then variety data from the product and process levels will be break down into individual product items and process elements. In this way, diverse product items and process elements contained in all product and routing variants are organized one by one in systematic data tables. These data tables all together forms a Transaction Database (TDB) which consists of two sets of records: product variety instances, $X=\{V_i^*\}$ and process variety instances $Y=\{a_j\}\vee\{M_k\}\vee\{T_s\}\vee\{F_t\}$, where a_j is process type instances, M_k is process machine instances, T_s is process tooling instances and F_t is process fixture instances. Then the support level ($s\%$) and confidence level ($c\%$) can be calculated by the following equations:

$$s\%=\frac{Count(X\wedge Y)}{Count(TDB)}\times 100\% \;; \qquad (5.1)$$

$$c\%=\frac{Conut(X\wedge Y)}{Count(X)}\times 100\% \;. \qquad (5.2)$$

An association rule hence means that the occurrence of certain variety parameter instances of product variety is correlated to the occurrence of certain instances of process elements with s% support and c% confidence. These associations are in low-level and cannot explain variety mapping at product family level. It is thus necessary to generalize individual rules as generic associations at the class levels. Each generic relationship $\Psi(I^G, O^G)$ relates a generic product item I^G to a set of generic process items O^G .The result of rule generalization is a list of generic association rules between generic products and certain generic process elements, along with selection conditions for each generic rule and its occurrence frequencies.

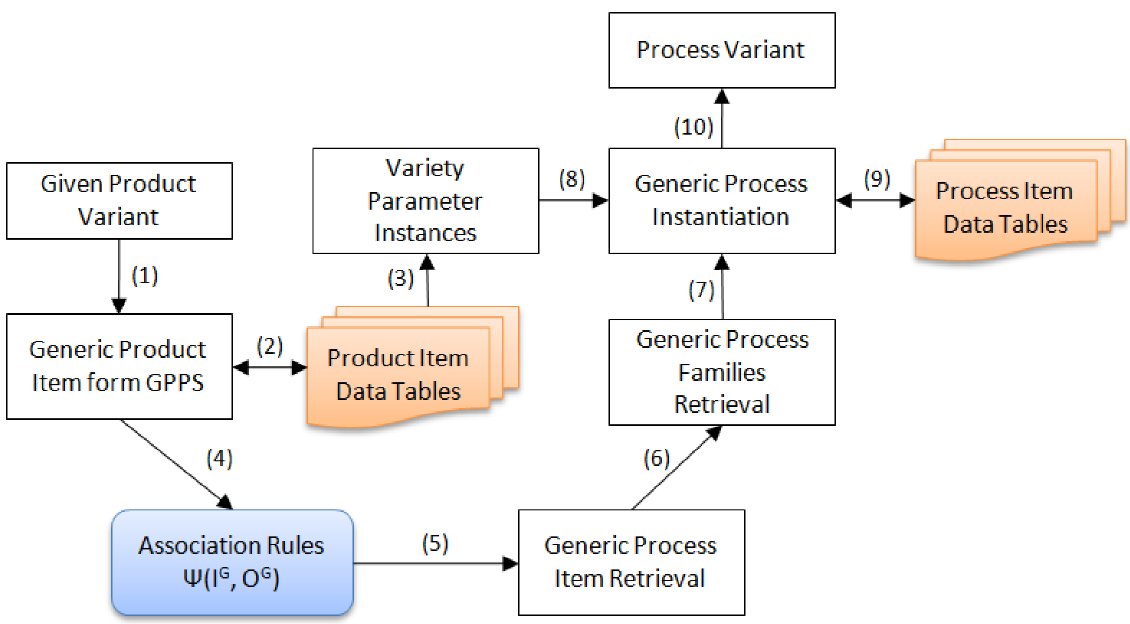


Figure 6.8 Process variant mapping from product variants based on association rules

The associations identified for product and process variety mapping lend themselves to be a mechanism to identify related process variants from given product variants. Figure 6.8 illustrates the general process of deriving a specific process variant from a given product variant. The steps as follows:

Step 1: The particular product variant is broken down into product items and then identifies its generic product item from the GPPS;

Step 2: The product items are also located in the data table;

Step 3: The unique variety parameter instances for each product item are identified;

Step 4: Based on the generic product item, the association rule base is explored;

Step 5: A relevant generic process item is identified from the association rule;

Step 6: Then the generic process elements are augmented and mapped to one of the existing process families;

Step 7 & 8: A specific process instance is thus derived by applying the unique set of variety parameter instances to this generic process;

Step 9 & 10: This process variant is matched with the process items data table in order to compose the specific process plan corresponding to the given product variant.

Likewise, for more product variants of the product family, this procedure can be applied, so as to identify all the related process variants for process planning within the company's existing manufacturing capabilities.

6.6 Discussions

Most of today's manufacturing companies will keep a product and process database which contains tons of data. But those data are mostly disconnected with each other which bring many difficulties when identification of certain process variety from a related process plan is needed. In high variety assembly system design, the first task is to identify the assembly processes available to achieve the targeted product variety. The mapping between product variety and assembly process as a variety enabler becomes the fundamental problem to the assembly system design task.

However, before identifying the mapping relationship, the GPPS is introduced to allow more efficient data handling. The GPPS provides a generic data structure to store and present product and process variants based on object oriented model. In the context of high variety assembly systems, it becomes an essential data structure as the large amount of variety data would cause very high computation load when handling the association rules mining tasks. The association rules mining also benefits from GPPS as the generic features allow the rules identification to be more reliable.

6.7 Summary

The original achievements in this chapter can be concluded as followings:

1) The generic product and process structure is suitable to store and represent product and process variety. Being a variety-oriented data structure, it allows higher efficiency on high variety handling and allows rule mining between product and process variety;
2) The association rule mining method can be used to identify the relationship of generic product variety and process variety;
3) The process flow proposed is capable of delivering process variety propagation which convert customer demands to corresponding assembly process sets.

Following the steps from the derivation of GPPS of a product family, product variety propagation, and then identify the related process variants based on the mapping between product and process variety, the first building block of high variety assembly system design is developed. By mapping from a product family design solution to process variety and its related assembly process items, a set of available assembly processes can be identified. It is the fundamental of high variety assembly system design problem.

CHAPTER 7

ASSEMBLY PROCESS DESIGN EVALUATION

7.1 Introduction

High variety and low volumes are the key challenges for efficient fulfillment of flexible assembly systems, in which optimal assembly process design and planning are critical. Assembly process design aims to determine appropriate manufacturing methods (called processes, or commonly referred to as processing jobs) for each individual part or sub-assembly according to its functional and structural characteristics. Subsequently, assembly process planning deals with how to implement multiple assembly process designs by utilizing extant assembly capabilities and resources. It involves optimal resource allocation and assembly process sequencing in order to ensure fulfillment and on-time delivery of all the jobs. High variety product would normally lead to high variety of assembly processes, which is not desirable in a traditional assembly system. A larger number of different assembly process instances will require more different types of assembly stations / work centers, thus increase capital cost, machine floor space and complexity of assembly process planning. It will also impact the utilization rate of individual assembly stations / work centers as less assembly resources are reused in this case. Assembly process reuse and commonality suggests itself to be an important instrument to leverage various manufacturing resources, enhance utilization rates and thus reduce costs on high variety product assembly.

In order to find the optimal assembly process design based on a given available assembly process set derived from Chapter 6, to identify an effective assembly process design evaluation become a determinant part in assembly system design. Generally speaking, the high variety assembly system design as an enabler of mass customization, it have two optimal goals, which are high variety and low cost. The key to high variety and low cost relies to flexible assembly processes, assembly process reuse, and high resource utilization. In this chapter, I identified two major assembly process evaluation criteria and their derivation. These evaluation criteria will be used as optimization goals in the case study of game theoretic optimization for optimal assembly process design and planning. As being identified in Chapter 3.3, the proposed methodology to solve the research problem includes:

1) Use matrix of assembly process ID and assembly resource ID to represent the assembly process design and resource allocation plan;
2) Define the assembly process commonality and resource utilization based on factors such as setup time, processing time, process and resource priority rating, demand etc., which is feasible data in assembly system design and independent to the product's geometry.

7.2 Evaluation Criteria in Process Design

Given multiple schemes of assembly process design, an optimal alternative could be selected with respect to different sets of evaluation criteria. For example, an appropriate assembly process plan can be decided according to status of recourse utilization (Tonshoff et al., 1989), by shortest processing time (Kim and Egbelu, 1999; Jian et al., 2006; Leung et al., 2010), in light of minimal tardiness (Weintraub et al., 1999), in terms of maximal diversity of equipment engaged to fulfill all the job (Saygin and Kilic, 1999), or based on whichever achieves the lowest manufacturing cost (Wang et al., 2008; Haddadzade and Farahnakian, 2009).

When implementing product family design, assembly process commonality becomes an important performance index of process reuse. Treleven et al. (1987) summarize the sources of process commonality by three categories: (a) reduction of setup time at each assembly process (an ideal case should be 0), (b) more flexibility in switching among jobs, and (c) more flexibility in adjusting a production plan in the short term. Accordingly, three respective types of commonality indices are developed for three different commonality sources. In the high variety assembly systems, the most essential goal is to reduce setup times across different assembly jobs, so as to enable small batch sizes economically and facilitate flexibility in production scheduling. Tsubone et al.(1994) discuss the impact of part commonality and process commonality on manufacturing systems and develop a process commonality index, $F = \sum_{h=1}^{H}\sum_{k=1}^{M} v_{h,k} / M$, where H is the total number of assembly process jobs, $v_{h,k}$ is a binary variable, assuming 1 if part k can be assembled by process h; otherwise is 0. Jiao and Tseng (2000) comprehend Treleven and Tsubone's (1987) process indices (PC_{ls} , PC_{sf} , PC_{ef}) and develop a process commonality index ($CI^{(P)}$) that taking into accounting common assembly processes, setups, lot sizing, and process flexibility. Kota et al. (2000) propose a product line commonality index (*PCI*) that evaluates commonality based on the parts' geometric dimensions, materials, manufacturing processes, and assembly procedures. Thevenot and Simpson (2006) extend *PCI* to include commonly shared parts, part volume and cost, leading to a comprehensive metric commonality (*CMC*). Table 7.1 draws a comparison of leading commonality indices in the literature.

Table 7.1State-of-the-art process commonality indices

Process Commonality Index	PC_{ls} , PC_{sf} , PC_{ef}	F	$CI^{(P)}$	PCI	CMC
Author	Treleven and Wacker(1987)	Tsubone et al.(1994)	Jiao and Tseng(2000)	Kota et al.(2000)	Thevenot and Simpson(2006)
Factors	Scale Economy Part Schedule Flexibility Expediting	Part Process Similarity	Scale Economy Part Schedule Flexibility Production Schedule Flexibility Part Process Similarity	Part Shape Difference Part Size Difference Part Material Difference Part Process Difference Part Assembly Difference	Part Shape Difference Part Size Difference Part Material Difference Part Process Difference Part Assembly Difference Shared Parts Quantity Cost

As shown in Table 7.1, *PCI* and *CMC* involve analysis of differences in part shape and dimensions, which could result in tremendous computational burden. It is also doubtable in the evaluation of assembly parts, which sometimes do not have a clear relationship between their geometries and process commonalities. Both indices neglect analysis of setup, which is an important source of process commonality in assembly systems. Treleven and Wacker (1987) define PC_{ls} , PC_{sf} and PC_{ef} to characterize commonality originated from lot sizing, sequencing flexibility and expediting (stopping a part's assembly process and switch to another one), respectively. All these are closely correlated with setup time, as a shorter setup time can facilitate a smaller economy batch size. As a result, the setup time of a part with a large volume should contribute more to process commonality. In addition, process sequencing affects setup time, and in turn should affect process commonality as well (Kim and Bobrowski, 1994). The *F* index examines similarity of process designs, as a similar routing plan means handling more part jobs with less machine tools to be used. Therefore, routing similarity should be incorporated into commonality analysis. Jiao and Tseng (2000) formulate a more comprehensive $CI^{(p)}$ based on PCI_{LS}, PCI_{SF} and F. It is generally characterized by the mean utilization of manufacturing capabilities for producing all the internally made parts and end products in the family.

7.3 Assembly Process Representation

Before evaluation criteria formulation, it is important to setup a standard assembly process mathematical representation. The process variety derivation resulted from product variety mapping can be used to match with the process items data table from the company's process database. The composing result would be an available process plan set corresponding to the given product variant. The set contains information including assembly process type, machine type, tooling, fixture and assembly cycle time.

7.3.1 Representation of Assembly Process Plan

Assume there are n_p types of processes, which forms an assembly process set P. Consider there are a maximal number of u processes contained in the assembly process plan, which can be presented as a vector $x_{ia} = (x_{ia1}, x_{ia2}, \cdots x_{iaj}, \cdots x_{iau})^T$, where x_{ia} is the a-th process plan of part i, and $1 \le a \le \Omega_i$. Ω_i is the total number of possible assembly process plans of part i, and $x_{iaj} \in P$ is the process code of process j in the a-th process plan of part i. If the number of the processes used in an assembly process plan is less than

u, dummy processes can be used in the generic representation of vector x_{ia} by assigning 0 as the process code.

Consider a maximal number of w feasible assembly process plans are associated with a product family, i.e., $w = \prod_{i=1}^{n_d} \Omega_i$. In practice, the multiple assembly process plans for a part are equivalent, and there is one and only one out of w process plans that can be selected for the part. The r-th ($1 \le r \le w$) assembly process plan set can be delineated in a matrix form, i.e.,

$$\mathbf{X}_r = \left(x_{1r}, x_{2r}, \cdots x_{ir}, \cdots x_{n_d r} \right) = \begin{pmatrix} x_{1r1} & \cdots & x_{n_d r1} \\ \vdots & \ddots & \vdots \\ x_{1ru} & \cdots & x_{n_d ru} \end{pmatrix}, \tag{7.1}$$

where each vector corresponds to a specific assembly process plan of the part, and n_d is the total number of parts in the product family.

7.3.2 Representation of Resource Allocation

Each resource type is associated with certain constraints, for which assembly process planning is supposed to allocate resources appropriately among multiple assembly jobs subject to a company's production capacities. It is a common practice for a company to maintain more than one instance of any particular resource type. Let $n_l \ge 1$ denote the number of instances for the l-th resource type ($1 \le l \le L$). Theoretically, a number of L type of resources can be grouped as a number of $G = \prod_{l=1}^{L} (n_l + 1) - 1$ resource clusters, such that a resource cluster, $g_m = (\tilde{g}_{1m}, \cdots \tilde{g}_{lm}, \cdots \tilde{g}_{Lm})$, $1 \le m \le G$ and $\tilde{g}_{lm}$ suggests the l-th type of resource contained in resource cluster g_m. If a resource cluster does not need a type of resource, the corresponding index $\tilde{g}_{lm} = 0$.

An assembly process plan, $\mathbf{X}_r$ $(1 \le r \le w)$ assumes a number of $v \in N^+$ feasible resource allocation plans. The s-th resource allocation plan $(1 \le s \le v)$ can be described as $\mathbf{Y}_{rs} = \left(y_{rs1}, y_{rs2}, \cdots y_{rsi}, \cdots y_{rsn_d}\right)$, where $y_{rsi} = \left(y_{rsi1}, y_{rsi2}, \cdots y_{rsij}, \cdots y_{rsiu}\right)^{\mathrm{T}}$ and y_{rsi} indicates the s-th resource allocation plan of the i-th part, such that y_{rsij} indicates the resource cluster of the j-th process. Finally a number of w process plans constitute a $w \times v$ resource allocation plan set Y in the matrix form, i.e.,

$$\mathbf{Y} = \begin{pmatrix} \mathbf{Y}_{11} & \cdots & \mathbf{Y}_{1s} & \cdots & \mathbf{Y}_{1v} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{Y}_{r1} & \cdots & \mathbf{Y}_{rs} & \cdots & \mathbf{Y}_{rv} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{Y}_{w1} & \cdots & \mathbf{Y}_{ws} & \cdots & \mathbf{Y}_{wv} \end{pmatrix}, \tag{7.2}$$

where each row of $\mathbf{Y}$ describes the entire set of resource allocation plans for an assembly process plan. Matrix $\mathbf{Y}$ contains a number of v columns, which consist with the maximal number of resource allocation plans among all the process plans. If an assembly process plan entails a less than v number of actual resource allocation plans, dummy resource allocation plans should be deployed in order to compose $\mathbf{Y}$ by assigning the respective matrix element values as 0. For example, if there are only two assembly process plans, $\mathbf{X}_1$ and $\mathbf{X}_2$, where $\mathbf{X}_1$ assumes three resource allocation matrixes, $\mathbf{Y}_{11}$, $\mathbf{Y}_{12}$, and $\mathbf{Y}_{13}$, and $\mathbf{X}_2$ contains two: $\mathbf{Y}_{21}$ and $\mathbf{Y}_{22}$. Hence $\mathbf{X} = \left(\mathbf{X}_1, \mathbf{X}_2\right)$ should entails a resource allocation matrix, $\mathbf{Y}=\begin{pmatrix} \mathbf{Y}_{11} & \mathbf{Y}_{12} & \mathbf{Y}_{13} \\ \mathbf{Y}_{21} & \mathbf{Y}_{22} & \mathbf{0} \end{pmatrix}$,in which the second row represents all there source allocation matrixes of $\mathbf{X}_2$ with one dummy matrix, $\mathbf{Y}_{23}=0$.

7.4 Formulation of Assembly Process Commonality

Evaluation of assembly process reuse and commonality generally involves three factors: (a) assembly flexibility in terms of lot sizes, magnitude of setup times, and number of alternate assembly methods; (b) sequencing flexibility of the processing jobs at each workstation; and (c) feasibility of assembly process selection in order to leverage extant assembly methods while prioritizing advanced flexible assembly technologies. Accordingly, this research defines three types of process commonality indices: (a) process flexibility index *C*, with reference to $CI^{(P)}$ by Jiao and Tseng (2000); (b) process sequencing flexibility index *F*, extended from PC_{sf} by Treleven and Wacker (1987); and (c) process feasibility index *A*.

7.4.1 Setup Time Matrix and Optimal Setup Time Table

Before the formulation of the assembly process commonality indices, a setup time matrix and optimal setup table is needed. Setup time refers to the time needed for a job to proceed from end of one assembly process to start of the subsequent assembly along the process route. It is also equivalent to the time of one resource switching to another. Setup time is only related to the assembly method employed, instead of resources types (Treleven, 1987), that is, relevant to $\mathbf{X}$, but not $\mathbf{Y}$. For computational convenience, I differentiate setup time by two portions: SET^1 and SET^2. Setup time SET^1 is the total time of the resources in a resource cluster changing from an initial status to an active in use status, for example, the total time of setting up machines, fixtures and tooling. The second portion, SET^2, is the total time of every resource changing its status from active to the initial inactive status, such as unloading of tools, fixtures, and cleaning up of equipment. The quantity of SET^1 and SET^2 depends on the bottleneck resource in the cluster. Since one part's assembly job takes place in a specific workstations, the numbers of parts and workstations are used as the unit measure of setup times.

(1) Setup Time Matrix: Given a number of n_e workstations in a company, a number of n_e matrixes of setup times can be produced at the workstations. For a resource allocation plan, $y \in \mathbf{Y}$, the total number of processes at the *e*-th $(1 \le e \le n_e)$ workstation should be u', corresponding to a $u' \times u'$ setup time matrix, $\mathbf{H}_{ye}$, as the following:

$$\mathbf{H}_{ye} = \begin{pmatrix} (\xi_{ye})_{11} & \cdots & (\xi_{ye})_{1k} & \cdots & (\xi_{ye})_{1u'} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ (\xi_{ye})_{h1} & \cdots & (\xi_{ye})_{hk} & \cdots & (\xi_{ye})_{hu'} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ (\xi_{ye})_{u'1} & \cdots & (\xi_{ye})_{u'k} & \cdots & (\xi_{ye})_{u'u'} \end{pmatrix}, \tag{7.3}$$

where each matrix element, $(\xi_{ye})_{hk}$, denotes the setup time for the *h*-th row's process (i.e., part *i* at process *j*) switching to the *k*-th column's process(i.e., part *o* at process *q*). It can be calculated as:

$$(\xi_{ye})_{hk} = \begin{cases} SET_{ij}^2 + SET_{oq}^1 & i = o,\ q = j+1 \\ 0 & i = o,\ q = j \\ \infty & i = o,\ q \ne j+1 \vee q \ne j \\ SET_{ij}^2 + SET_{oq}^1 & i \ne o \end{cases}. \tag{7.4}$$

For illustrative simplicity, a setup time matrix at a workstation could be formed by arranging the rows and columns in an ascending order of part IDs. If multiple processes are designed for a part at one workstation, these processes can be arranged by the order of its assembly sequence in a route.

(2) Optimal Setup Time Table: Calculation of the total setup time for an optimal assembly process plan can be derived from an optimal setup time table with respect to each individual part. There is one optimal setup time table corresponding to every process plan, in which each element of the table, t_{ij}^{SET}, indicates the setup time of a part's assembly process in an optimal route. For any assembly process plan $x \in \mathbf{X}$, the corresponding resource allocation plan should suggest at least one optimal processing sequence, z^*, such that $t_{ij}^{SET} = SET_{oh}^2 + SET_{ij}^1$, if the *j*-th process of part *i* takes place succeeding to the *q*-th process of part *o*; and $t_{ij}^{SET} = SET_{ij}^1 + SET_{ij}^2$ if the process is supposed to begin as the first one.

7.4.2 Assembly Flexibility Index

Jiao and Tseng (2000) originally identified the process commonality of product family by the mean utilization of manufacturing capabilities for producing all the internally made parts and end products in the family. It is defined as follows:

$$CI_{(1)}^{(P)} = \frac{\sum_{p=1}^{n_p} \sum_{i=1}^{n_d} \lambda i_p}{n_p}, 1 \le CI_{(1)}^{(P)} \le \beta_{(1)} \tag{7.5}$$

where λ_{ip} indicates whether part *i* employs assembly process *p* ($\lambda_{ip} = 1$), or not ($\lambda_{ip} = 0$). While process flexibility represents an important dimension of process commonality measurements, lot sizing is another area in which process commonality has been recognized as having an impact on assembly system design and planning. In determining the appropriate economic lot size, the set-up time (cost) required is a key factor since lower set-up times result in smaller economic lot sizes. Therefore, the appropriate measure of this lot sizing component of process commonality should consider the mean of the set-up times for all possible set-ups at a particular process (Treleven and Wacker, 1987). Less average assembly process time needed for a part also imply more similarity of the assembly jobs among parts, and thus facilitate assembly process flexibility. Likewise, large volume of components means more impact of its setup time on assembly process commonality. Based on $CI^{(p)}$ by Jiao and Tseng (2000), by adding factors of setup time and processing time to Eq.(7.5), an assembly process flexibility index for process plan $x \in X$ can be defined as the following:

$$C(x) = \frac{\sum_{p=1}^{n_p}\sum_{i=1}^{n_d}\lambda_{ixp}T^{SET}T^{p}\sum_{i=1}^{n_d}D_i}{\sum_{p=1}^{n_p}\eta_p\sum_{i=1}^{n_d}\left(D_i\sum_{j=1}^{u}(t_{ij}^{SET}\cdot t_{ij}^{p})\right)}, \tag{7.6}$$

where λ_{ixp}: Indicating whether part i employs assembly process p in process plan x ($\lambda_{ixp}=1$), or not ($\lambda_{ixp}=0$);

T^{SET}: Total setup time of all the parts when organizing production assembly to the shortest setup time principle, which is calculated by Eq. (7.10);

T^{p}: Total process time of all the parts when on given process design and plan;

D_i: Volume of demand for part i;

$\sum_{p=1}^{n_p}\eta_p$: Total number of processing methods in process plan x, where

$$\eta_p = \begin{cases} 1 & if\ \forall x_{iaj}\in x,\ x_{iaj}-p=0, i\in\{1,2,\cdots,n_d\}, a\in\Omega_i, j\in\{1,2,\cdots,u\} \\ 0 & if\ \forall x_{iaj}\in x,\ x_{iaj}-p\neq 0, i\in\{1,2,\cdots,n_d\}, a\in\Omega_i, j\in\{1,2,\cdots,u\} \end{cases};$$

t_{ij}^{SET} : Derived from the setup time table and indicating the setup time of process *j* for part *i*;

t_{ij}^{p} : Derived from the process time table and indicating the assembly process time of process *j* for part *i*.

7.4.3 Sequencing Flexibility Index

Assembly sequencing flexibility refers to difference in the magnitude of setup times among different assembly sequencing plans across multiple workstations. More flexibility in assembly sequencing implies indifference of diverse assembly sequences in terms of setup times. In other words, having set-up times that are sequence independent allows the scheduler greater freedom in determining the order in which the jobs are to be scheduled. Derived from PC_{sf} by Treleven and Wacker (1987), this research defines a sequencing flexibility index for resource allocation plan $y \in Y$ as the following:

$$F(y) = 1 + \sum_{e=1}^{n_e} \left(\frac{\sum_{h=1}^{u'} \sum_{k=1}^{u'} \left((\xi_{ye})_{hk} (\alpha_{ye})_{hk} - \frac{\sum_{i=1}^{u'} \sum_{j=1}^{u'} ((\xi_{ye})_{ij} (\alpha_{ye})_{ij})}{\sum_{i=1}^{u'} \sum_{j=1}^{u'} (\alpha_{ye})_{ij}} \right)^2}{\sum_{i=1}^{u'} \sum_{j=1}^{u'} (\alpha_{ye})_{ij}} \right), \qquad (7.7)$$

where ξ_{ye} is setup time defined in Eq. (7.4), $(\alpha_{ye})_{hk} = \begin{cases} 0 & (\xi_{ye})_{hk} = 0 \text{ or } \infty \\ 1 & (\xi_{ye})_{hk} \neq 0 \wedge (\xi_{ye})_{hk} \neq \infty \end{cases}$, and

$$(\alpha_{ye})_{ij} = \begin{cases} 0 & (\xi_{ye})_{ij} = 0 \text{ or } \infty \\ 1 & (\xi_{ye})_{ij} \neq 0 \wedge (\xi_{ye})_{ij} \neq \infty \end{cases}.$$

In Eq. (7.7), the maximum value of $F(y)$ is 1, when the setup times required for all jobs at a workstation are identical.

7.4.4 Assembly Feasibility Index

Assembly process planning must consider the feasibility of an assembly process plan in such a way that a part with higher priority should be assigned with more efficient assembly method. Nonetheless every part by itself intends to be planned with more advanced assembly methods. However the capacities of the needed resources of each method are limited in terms of assembly time available for each type of resources. Therefore, assembly feasibility is an important measurement for allocating advanced assembly methods to those parts with higher priorities. This measurement encourages each part to be planned with an assembly method that is consistent with the priority of the part. If no appropriate method matching with the priority, it suggests another advanced method with the motivation to enhancing the overall utilization of advanced assembly methods. For an assembly process plan, $x \in X$, its process feasibility can be defined as an index:

$$A(x,y) = 1 - \frac{\sum_{i=1}^{n_d} \sum_{j=1}^{u} K \left(L_i^p - L_{ij}^m \right)^2}{\sum_{i=1}^{n_d} (L_i^p)^2 + \sum_{i=1}^{n_d} \sum_{j=1}^{u} (L_{ij}^m)^2} \quad , \tag{7.8}$$

where L^p denotes the priority of a part in terms of complexity of part design, cycle time, cost, tolerance requirements, etc.; and L^m indicates the priority of an assembly method according to its efficiency, accuracy, etc. Part priority L^p and processing method priority L^m are determined by three levels from low to high, which are denoted as 3, 2, and 1, respectively. To enhance utilization of advanced assembly methods in a firm, Eq. (7.8) adopts a coefficient K to reflect the practical situation. For instance, K =0.02 when $L^p \leq L^m$; or when $L^p > L^m$, K =0.01. An ideal case is $L^p = L^m$, when the priorities of the parts and assembly methods are all matched, and thus A equals 1. Except this ideal case, the value of A is always less than 1, i.e., $A \in$ [0,1), reflecting the priority of a assembly method should match with that of a job in assembly process design. In case when no restriction on an advanced processing method to be employed for a low-priority part, we can allow K =0 for the case when $L^p > L^m$.

7.5 Evaluation of Resource Utilization

7.5.1 Resource Utilization Rate

The utilization rate of a particular resource is commonly measured according to the ratio of the resource's busy time (i.e., processing time plus setup time) to its availability in terms of total available time. A higher rate means more effectiveness of resource utilization with less portion of idle time (i.e., total available time less busy time). Let T_e^p denote the processing time of all the jobs assigned to workstation e and T_e^a stand for the total available time of workstation *e*. If let τ_{ij} indicate the assembly time of process *j* of part *i* at workstation *e*, the assembly time can be expressed as the following:

$$T_e^p = \sum_{i=1}^{n_d} \sum_{j=1}^{u} \gamma_{ij} \tau_{ij}, \qquad \gamma_{ij} = \begin{cases} 1 & \text{process } j \text{ of part } i \text{ is allocated to workstation } e \\ 0 & \text{process } j \text{ of part } i \text{ is not allocated to workstation } e \end{cases}$$

Then let T_e^{SET} delineate the total setup time at workstation *e*, the basic measure of resource utilization at workstation *e* can be denoted as $\left(T_e^p + T_e^{SET}\right) / T_e^a$.

However, the basic measurement above can hardly differentiate the impact of assembly time T_e^p and setup time T_e^{SET} on the overall utilization, as they are accounted for as one aggregated amount of busy time. In practice, out of a particular period of busy time, a larger portion of assembly time and less portion of setup time (i.e., a larger T_e^p / T_e^{SET} value) indeed suggest that there is more time for effective assembly process, and thus should contribute to a higher utilization rate, even though the amounts of total available time and idle time keep intact. I propose to incorporate such an impact of process effectiveness, T_e^p / T_e^{SET}, into the basic resource utilization measurement through a scaling factor in the form of an exponential recovery function, i.e., $\Gamma = 1 - e^{-\psi T_e^p / T_e^{SET}}$,where $\psi \in (0,1]$ performs as a sensitivity constant of the T_e^p / T_e^{SET} impact on the utilization rate. A smaller value of ψ reflects more sensitive of the impact. The actual value is determined according to practical operations of the assembly system, for example by benchmarking with well-established standards of work measurement and time studies in a company (Jiao and Tseng, 1999). Note that $0 < \Gamma < 1$ and it is enacted as a scaling factor that is nonlinearly proportional to the process effectiveness reflected in T_e^p / T_e^{SET}.

To incorporate the above two aspects of resource utilization, we can define a comprehensive utilization rate for a resource allocation plan $y \in Y$ as the following:

$$E(x, y) = \Gamma \left(\frac{\sum_{e=1}^{n_e} T_e^p + \sum_{e=1}^{n_e} T_e^{SET}}{\sum_{e=1}^{n_e} T_e^a} \right) = \left(1 - \exp\left(-\psi \left(\frac{\sum_{e=1}^{n_e} T_e^p}{\sum_{e=1}^{n_e} T_e^{SET}} \right) \right) \right) \left(\frac{\sum_{e=1}^{n_e} T_e^p + \sum_{e=1}^{n_e} T_e^{SET}}{\sum_{e=1}^{n_e} T_e^a} \right) \quad (7.9)$$

The value of $E(x, y)$ ranges from 0 to 1, while a larger value indicates a higher utilization rate. It is accounted by assessing all resources engaged in the workstations that are employed in a resource allocation plan $y \in Y$.

7.5.2 Calculation of Setup Time

To improve assembly process commonality within a product family, assembly process planning should emphasize minimization of total setup time. Assembly sequencing at each workstation in turn should abide by a shortest setup time principle. To calculate the total setup time at workstation e, I construct a square matrix $\mathbf{B}_{ye}$ that is of the same order as setup time matrix $\mathbf{H}_{ye}$, in which a matrix element, $(\beta_{ye})_{hk}$, corresponds to the h-th row and k-th column, assuming a value of 1 or 0. A procedure of constructing matrix $\mathbf{B}_{ye}$ is carried out in the following steps:

Step 1: Duplicate the setup time matrix $\mathbf{H}_{ye}$ at workstation e to $\mathbf{B}_{ye}$ and initiate all elements of $\mathbf{B}_{ye}$ to be 0;

Step 2: Randomly generate a sequence for all the jobs, yielding a total number of $n_z = u'!$ assembly sequencing plans;

Step 3: Choose any sequencing plan z ($1 \le z \le n_z$) to judge the value of element $(\beta_{ye})^z_{hk}$ in matrix $\mathbf{B}^z_{ye}$ according to: If process *j* of part *i* is prior to process *q* of part *o* in assembly sequence *z*, then set element $(\beta_{ye})^z_{hk}$ in matrix $\mathbf{B}_{ye}$ to 1. Note process *j* of part *i* corresponds to row *h*, whilst process *q* of part *o* the *k*-th column;

Step 4: Repeat Step 3, leading to a number of n_z respective square matrix $\mathbf{B}_{ye}$ for a number of n_z assembly sequences;

Step 5: Among a number of n_z sequences, find one with the shortest setup time, z^*,whose square matrix is $\mathbf{B}^{z^*}_{ye}$.

Once matrix $\mathbf{B}_{ye}$ is known, the shortest setup time among all the workstations can be calculated for any resource allocation plan, $y \in \mathbf{Y}$, as the following:

$$T^{SET} = \sum_{e=1}^{n_e} \left(\sum_{h=1}^{u'} \sum_{k=1}^{u'} (\beta_{ye})_{hk}^{z^*} \ (\xi_{ye})_{hk} + S_{FL} \right). \tag{7.10}$$

The total setup time at workstation e is determined as:

$$T_e^{SET} = \sum_{h=1}^{u'} \sum_{k=1}^{u'} (\beta_{ye})_{hk}^{z^*} \ (\xi_{ye})_{hk} + S_{FL} \ , \tag{7.11}$$

where $(\beta_{ye})_{hk}^{z^*}$ is matrix element of $B_{ye}^{z^*}$; and for an optimal sequence plan z^*, S_{FL} equals the SET^1 of the first assembly process plus the SET^2 of the last assembly process in the sequence.

7.6 Discussions

The assembly process commonality introduced in this chapter involves assembly flexibility index, sequencing flexibility index and assembly feasibility index. All these evaluation criteria together will help to quantify the quality on a given selection of assembly process design. These three indices is a reflection on the major concerns in an assembly system design when high variety, high mix and low volume are needed. It is not much concerned in traditional assembly systems as most of them are just dealing with one or several fixed product design in high volume. In this case, change over time has very limited impact on system through put. And sequencing is also not an issue as they do not require change of assembly sequence to allow working on mixed production schedule. Assembly feasibility is a new challenge brought by flexible manufacturing systems as only such systems may have a situation of using the same manufacturing resource to produce different products. The new challenges come with high variety assembly systems: quick changeover, flexible sequencing and resource feasibility are fully addressed in the definition of assembly process commonality index.

Resource utilization rate is a very important factor to determine if the assembly process plan is efficient. As a higher utilization of manufacturing resources would eventually lead to a lower waste of manufacturing capability, which is the goal of a kaizen concept (Fieldbook, 2004), would significantly affect the cost of production.

7.7 Summary

The original achievements in this chapter can be concluded as followings:

1) The proposed assembly process commonality evaluation criteria reflects the flexibility of given process design, which involves process utilization, lot sizing factors, process sequencing flexibility and mixed newer and older assembly resources which is common in reality;
2) The proposed assembly resource utilization evaluation criteria reflects the efficiency of given process plan, which considers not only overall machine busy / free time ratio, but also the process effectiveness.

By introducing the assembly commonality index and resource utilization index, the evaluation of both assembly process design and assembly process planning will enable the LFGA based search for optimal assembly system design solution in a given design set. By setting the assembly commonality as fitness function of ULC and resource utilization as LLC, a game theoretic optimization of high variety assembly process design and planning would be formulated. Before the further discussion of the assembly process design and planning, the physical constraint of assembly system, system layout, would be discussed in the next chapter.

CHAPTER 8

ASSEMBLY SYSTEM LAYOUT DESIGN

An effective layout design upon a manufacturing system is significant (Canen and Williamson, 1996). In an assembly system, one of the major product cost is related to material handling. An efficient layout arrangement and material flow path design will benefit not only the factory floor space utilization, assembly process cost but also system through put and flexibility.

8.1 Introduction

Assembly system is a sequential organization of different manufacturing recourses, such as workers, tools or machines. In modern assembly systems, all parts or assemblies are handled either by conveyors or motorized vehicles with no manual trucking. The basic principles of assembly system are concluded as: (1) Each component part shall travel the least possible distance while in the process of assembly; (2) Use carrier so that after each operation, the part or subassembly stays at the same location and convenient for the next assembly process; (3) Use sliding assembling lines by which the parts to be assembled are delivered at convenient distances (Ford, 2007). The ultimate goal of an assembly system layout design is to minimize the total travel distance of material flow as well as maximize the factory floor space utilization. Traditional production line is a straight line with all assembly steps happen in a fixed order one by one, which is often used to take advantage of mass production. However, as a consequence of the implementation of mass customization principles into manufacturing, an assembly system which can handle multiple product variants has becoming more popular in industrial practice. Instead of traditional straight production lines, many companies are organizing their production processes into U shape, circular shape or other open-field shapes. Most of the assembly system shape layouts are fixed due to technological constraints. As the component, subassembly and final assembly lines are located within the same facility, the result is a complex system consisting of many different lines feeding one another. There are some flexible manufacturing system concepts applied in industry with modular design and interchangeable stations. But such kind of system layout reconfiguration still requires a major production down time and is only used when certain product family reaches its end of life and a new product family generation requires a reconfiguration. For a given product family with high product and assembly process variety, a fixed assembly system layout with flexible assembly units and process plan is still the best proven solution.

The layout design of manufacturing facilities is based on the concept of planning departments that are either process-based or product-based entities. The departments typically are constrained in terms of area. The detailed layout is assumed to follow the block layout design. As a result, the facility layout design formulation treats the departments as malleable objects, with department shape refinement to follow based on user-based massaging (Bukchin et al., 2006). The ever expanding robotic and flexible material handling technology enables modern assembly systems to handle high variety product without changing of assembly department layout design. It also allows multiple pick-up/drop-off (P/D) points in a certain department, which gives large numbers of freedom to assembly system department layout design.

Several facility planning techniques could be used to develop a new layout or improve the current layout such as Systematic Layout Planning (SLP), Pairwise Exchange Method (PEM), Graph Based Theory (GBT), Dimensionless Block Diagram (DBD), Total Closeness Rating (TCR), etc. (Ojaghi, 2015). SLP is a procedure developed by Muther (1973). It involves eleven steps and is able to find a number of solutions for the layout. Chien (1992) categorized the eleven steps of SLP into four parts that are data input, procedure's process, output results and evaluation process. Among these steps, the major task to identify the department interaction, then constructing a system layout to minimize the distance of inter-department material flow.

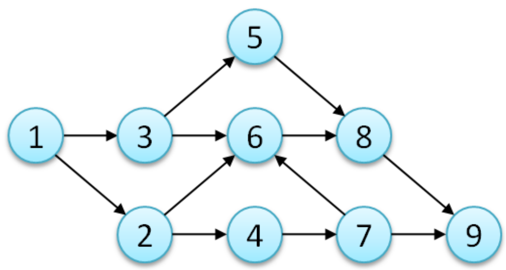
Clustering analysis is often used to group objects with stronger interactions, thus it is also widely used in SLP and GBT to determine the tendency of grouping among processes. Relationship chart (REL chart) is commonly used to determine the importance of adjacency between each pair of departments based on the designer's rating of TCR. However, in high variety assembly systems, the manufacturing resources are no longer needed to be grouped based on departments. Many of the high variety assembly machines such as robotic assembly stations, flexible feeders, and automatic kiting stations can be used in multiple locations and different stages during assembly, to group all similar work stations into departments could lead to inefficiency in a high variety assembly system. So the clustering analysis for individual assembly processes should be used instead of for departments. The REL chart which is based on TCR rating is limited to be used in department closeness rating, which has very limited number of entities to be analyzed. For assembly system with high process variety, a more efficient clustering analysis which is suitable for program controlled calculation is needed. In this chapter, I propose an assembly system layout deign method using design structure matrix. As being identified in Chapter 3.4, the proposed methodology to solve the research problem includes:

1) Use a graph based model to represent the interconnection between each assembly processes, which should be more rigorous than commonly used REL chart in industry and allows automatic Matlab calculation and optimization;
2) Use clustering analysis to identify the tendency of clustering of assembly processes in terms of overall system efficiency. The clusters will be used as a reference to build individual FMS cells.

8.2 Design Structure Matrix

The Design Structure Matrix (DSM - also known as the dependency structure matrix, dependency source matrix, and dependency structure method) is a general method for analyzing system models in a variety of application areas. A DSM is a square matrix (i.e., it has an equal number of rows and columns) that shows relationships between elements in a system (Hellenbrand and Lindemann, 2008). It is a powerful analyses method for clustering and sequencing problems. Since the behavior and value of many systems is largely determined by interactions between its constituent elements, DSMs have become increasingly useful and important in recent years.

There is no pre-defined DSM that is helpful for any problem that is to be structured. Rather, DSM needs to be adapted to the kinds of elements and relations that prevail in the system in focus. Basically, the type of the elements and dependencies needs to be defined as precisely as possible to obtain the information structure for the DSM. The activity-based DSM is defined to represent a set of tasks in a process. These tasks work together to fulfill the goal of the overall process. The exchange of information can thus be represented as a digraph or a DSM. The activity-based DSM can be used to represent any given assembly processes and through a series of matrix transformation, the interaction of each processes can be analyzed. As shown in Figure 8.1, an assembly process on the left can be converted to an activity-based DSM on the right.



	1	2	3	4	5	6	7	8	9
1		X	X						
2				X		X			
3					X	X			
4							X		
5								X	
6								X	
7						X			X
8									X
9									

Figure 8.1 Activity-based DSM conversion

The conversion can be automatically processed, as the mapping from the assembly process to the DSM is pre-defined. Each vector pointing from one process to the other is represented by one matching element in the DSM. The standard DSM is a binary matrix, which use 1 or "X" to represent the sequence's existence and 0 or empty to represent such sequence's non-existence.

8.3 DSM Clustering

When the DSM elements represent a set of assembly process, the goal of the matrix manipulation becomes finding subsets of DSM elements (i.e. clusters or modules) that are mutually exclusive or minimally interacting subsets, i.e. clusters as groups of elements that are interconnected among themselves to an important extent while being little connected to the rest of the system. This process is referred to as "Clustering". In other words, clusters absorb most, if not all, of the interactions internally and the interactions or links between separate clusters are eliminated or at least minimized.

In a modern assembly system utilizing robotic assembly and flexible material handling, the system geometric dimension is much smaller than the traditional fixed tooling assembly system or manual assembly lines. The highly integrated robotic assembly system together with machine vision and sensor technology makes a small foot print automatic assembly system possible. As a result, the internal material flow path of each automatic assembly work station is very short and highly efficient. By clustering a DSM which represents a set of assembly processes, the assembly processes can be divided into several departments where internal material flow is much more than external material flow. Then the total material flow distance can be minimized with most highly interactive assembly processes clustered into the same department or work station.

To optimize the arrangement of assembly process flows and find the most efficient clustering decision in DSM, a heuristic clustering algorithm is applied. The clustering algorithm is based on the work of Martín-Fernández (1998), where a coordination cost function can be developed to evaluate different clustering arrangements within the DSM. The coordination cost can be calculated by using the following equations:

If both process *i* and *j* are in a cluster *k*, then the coordination cost of process *i* is

$$CC(i) = \sum_{j=1}^{size} (DSM(i, j) + DSM(j, i)) \times \sum_{k=1}^{size} n_c(k)^p \;, \tag{8.1}$$

otherwise, if no cluster contains both *i* and *j*, the entire DSM acts as a cluster,

$$CC(i) = \sum_{j=1}^{size} (DSM(i, j) + DSM(j,i)) \times size^{p} \quad , \tag{8.2}$$

where *size* is the total size of the DSM, *DSM(i, j)* is the value of the interaction between process *i* and *j*, $n_c(k)$ is the number of processes contained in cluster *k*, *p* is the penalty assigned to the size of cluster in coordination cost. Then the overall coordination cost is the summation of coordination cost of all processes,

$$CC_{total} = \sum_{i=1}^{size} CC(i) \quad . \tag{8.3}$$

Initially, there are as many as clusters as there are DSM elements. The algorithm randomly selects an element and calculates a bid from clusters. The highest bid will be chosen, if the coordination cost will reduce, the element will be added into the cluster. This process will continue until the coordination cost converges.

8.4 Assembly System Layout Design using DSM

In assembly system layout design, a binary DSM is not enough to describe the nature of the assembly process, as different process flow will have different degree of influence to the overall material flow. A process correlation weight rating is needed to prioritize the clustering of those process flows with higher influence to the system. Sometimes, the assembly processes have different branches for different product variants, and different variants have different production volume, thus different impact to the total through put. Different assemble process flows would also have different weight because of sharing of tooling, machine or setups. This kind of process flows would have very strong tendency to stay within the same department or work station. Figure 8.2 shows an assembly process flow with weight rating.

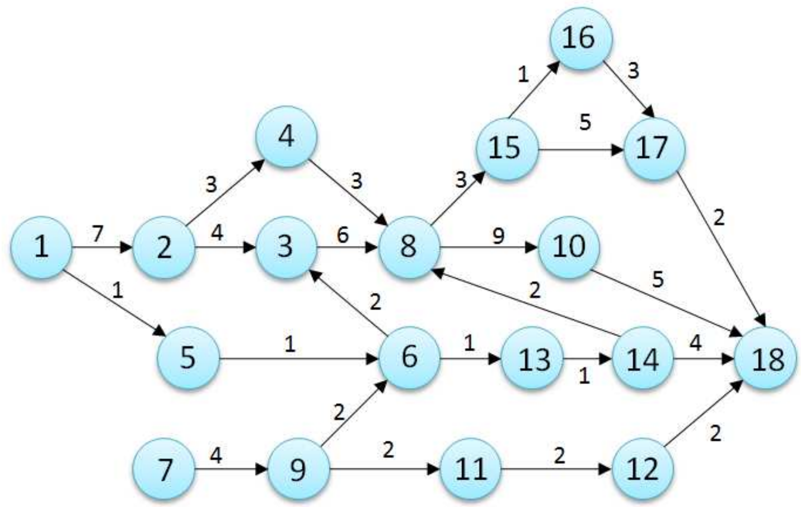


Figure 8.2 A weighted assembly process flow

By following the conversion rule of $DSM(i, j) = w(i, j)$, where $w(i, j)$ is the weight rating of process flow from *i* to *j*, the resulted DSM is shown in Figure 8.3.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1		7			1													
2			4	3														
3								6										
4								3										
5						1												
6			2										1					
7									4									
8										9					3			
9						2					2							
10																		5
11												2						
12																		2
13														1				
14								2										4
15																1	5	
16																	3	
17																		2
18																		

Figure 8.3 A weighted DSM of assembly process flow

After clustering analysis, the result is shown in Figure 8.4. There are seven clusters identified with a coordination cost of 690. The processes in each cluster are shown in Figure 8.5. Each cluster represented as a highlight box would be used to construct as an assembly department, which keeps the internal processes integrated as one work station and connects with other work station by external material flows.

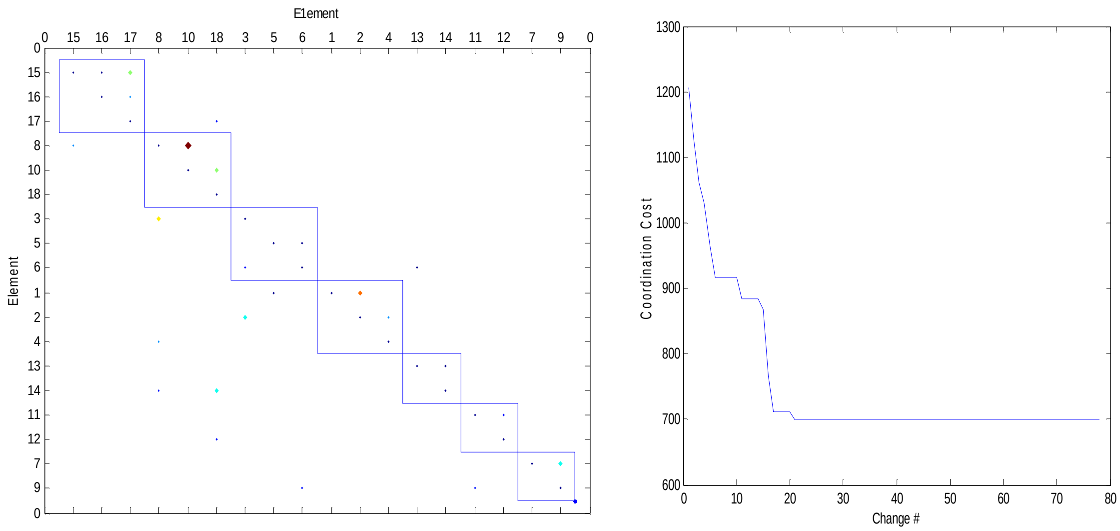


Figure 8.4 Clustering result using coordination cost function

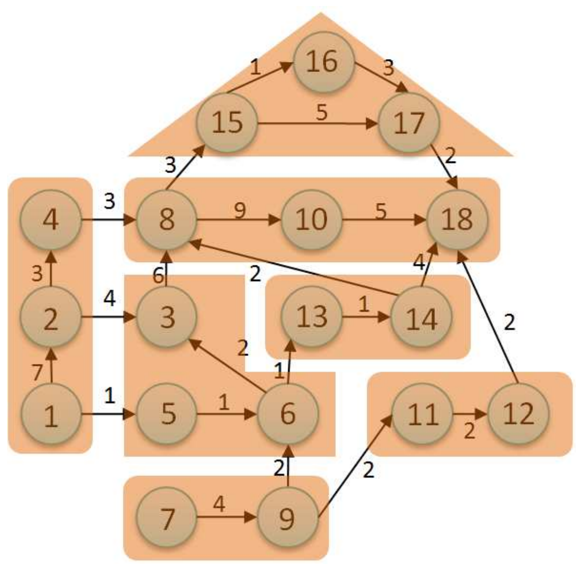


Figure 8.5 Clustering result of assembly processes

8.5 Discussions

The assembly system layout design is the physical foundation of the realization of high variety assembly system. The task of designing an assembly system layout can be generally concluded as grouping similar and highly dependent assembly processes together into an enclosed unit, a department or a work station, depending on their scale. Instead of using traditional graph based approaches or systematic layout planning, the DSM enables the whole layout design process to be automatic. The process flow conversion and matrix clustering can be carry out using generic software codes, which is independent of the assembly system itself. This is a very important feature to allow high variety production. In industrial applications, some high variety products with high mix and low quantity orders would have hundreds of different part numbers in a single product family. To deliver an assembly system layout for such assembly process sets would take a long time if the design is manually handled. Then it won't justify the cost as normally such kind of low volume high mix products will also have a short production cycle. DSM can represent all process sets using the same way as in GPPS, which allows the process data to be directly used in layout design. In real life applications, the penalty index p in the clustering process will need a DOE to identify the best value to achieve a reasonable converging speed as well as delivering a satisfactory result.

8.6 Summary

The original achievements in this chapter can be concluded as followings:

1) Instead of REL chart, the DSM is better to represent the relationship between assembly processes, which allows a more rigorous representation of the system interconnection and allows quantification and automatic data processing;
2) The coordination cost function and setup time / process time weighted assembly process is capable to cluster the high variety assembly process,

which is then used as reference of the physical assembly system layout design.

As a summary, I proposed a DSM based high variety assembly system layout design method, which takes the input form Chapter 6's process variety mining to deliver a layout design on given assembly process sets. It would be a physical constraint to the assembly process planning and input to the assembly system modeling. It can also serve as a reference to factory floor automatic assembly cells configuration. In the next chapter, an assembly system model capable of running online and response at real time is proposed.

CHAPTER 9

DATA DRIVEN SIMULATION OF ASSEMBLY SYSTEM PROCESS PLANNING

9.1 Introduction

The Robotic equipment has found great application to a broad range of automatic assembly systems, specifically in the assembly lines of automotive industry, electronics, rubber/plastics and metal/machinery industrial sectors. The robots' intrinsic characteristics, such as high accuracy, speed, repeatability, strength and reliability, have enabled production firms to invest in large scale installations that can work around the clock with minimal human intervention (Michalos et al., 2010). Nevertheless, technological limitations impose the contribution of human operators on the process, by providing support to the system (Šekoranja et al., 2014). The same as tooling and fixtures in both traditional assembly lines and modern flexible assembly systems, robotic equipments are considered as one of the major manufacturing resources. Given a list of candidate equipment available to complete the operations, the assembly process planning problem thus becomes to decide which resources to select and which tasks to assign to each resource in order to meet the production requirements at a minimum cost.

The utilization of robotic equipments and other flexible manufacturing resources also brings more and more sensing technologies into the high variety assembly system. The data exchange rate between individual assembly machines and control system has shown a huge incensement. Such data exchange capability shows great potential for utilizing the data generated from a running assembly system to help on the decision making on process planning. So a real-time data feedback and online simulation is proposed to take the advantage of modern flexible assembly systems for better process planning solution.

Many of the researchers have done Research on assembly lines process planning problem. However majority of the papers concentrates on ALB and development of different optimization algorithms. Research done in these areas have not considered stochastic nature of the various factors (i.e. changes in resources allocation, market demands, failures, delays, quality / tooling issues). In order to address these stochastic behaviors, a DES simulation model which is capable of simulating the system response to the resource allocation plan is necessary. In this chapter, a model driven assembly process planning method is proposed, it can also serve as a verification of assembly system layout design. The performance of an assembly system layout is always hard to analyze by analytical approaches in engineering practice. DES tools are used to assess such problems by modeling a running assembly system as a discrete sequence of events in time. Such model is conceptually built using ad-hoc methods by the simulation analysts based on their understanding of the system. These ad-hoc methods have no standards or structure to follow and can take various forms such as documents, diagrams, databases, etc.

Using a simulation language to represent the system impacts the fidelity of the communication between the domain engineers which may introduce doubt as to whether the simulation analysts have grasped fully their intent. In order to ensure that the simulation analysts receive the right information, significant time and effort are expended in this phase of any simulation project. Moreover, the informality of these methods hinders the re-usability of the system descriptions or investigating automatic model transformations (Batarseh and McGinnis, 2012). A more generalized language for system modeling which enables domain engineers with different background knowledge accessing is needed in the practice of DES. Such language can bridge the gap between domain engineers and simulation analysts and also should serves as a multi-disciplinary modeling tool which will allow cooperation of domain engineers of different sub-systems.

In this chapter, a data driven DES simulation is proposed for assembly system process planning. The simulation model can be directly translated from a generalized system model, which allows domain engineer's access to change running parameters and system configuration without knowledge of DES simulation language itself. Then, by providing a real-time data feedback to the simulation model, a data driven online simulation framework is proposed for high variety assembly system process planning. As being identified in Chapter 3.5, the proposed methodology to solve the research problem includes:

1) Identify a modeling method to formulate the simulation model using a intermediate domain knowledge independent system model;
2) Develop an assembly process planning method which is capable of monitoring and response to the highly dynamic high variety assembly system.

9.2 System modeling language

The System Modeling Language (SysML) is a general purpose modeling language for systems engineering applications and its scope goes through a wide range of systems, or systems of systems, including hardware, software, processes, facilities, etc. It is an extension of Unified Modeling Language (UML) designed to support systems engineering in general. As software generation from UML models is a common practice, it is reasonable to use SysML, which can be considered as a system modeling extension of UML, for assembly system layout modeling and simulation. SysML is widely used in system modeling science its introduction in 2006. The first trial of SysML model translation to DES is done by Huang et al. (2007) They successfully converts a standard three-unit machine shop model into Em-Plant. McGinnis et al. (2009) develop a SysML driven Arena model of the same machine shop example. A further study by this group uses the Atlas Transformation Language in Eclipse to do the translation work. Although the SysML model driven design and simulation has been studied for seven years, there are still not many papers on practical DES applications.

SysML is a graphical language which is easier to understand, in terms of its description of the system being simulated, than various DES tools. It is based on the XML Metadata Interchange (XMI) standard, which means it is object oriented and easily transformed “automatically” into its corresponding simulation model. By using SysML as an intermediate representation of the DES model, domain engineers can get involved into the modeling process and thus guarantee a coherent understanding in modeling. A simplified framework comparison is shown in Figure 9.1.

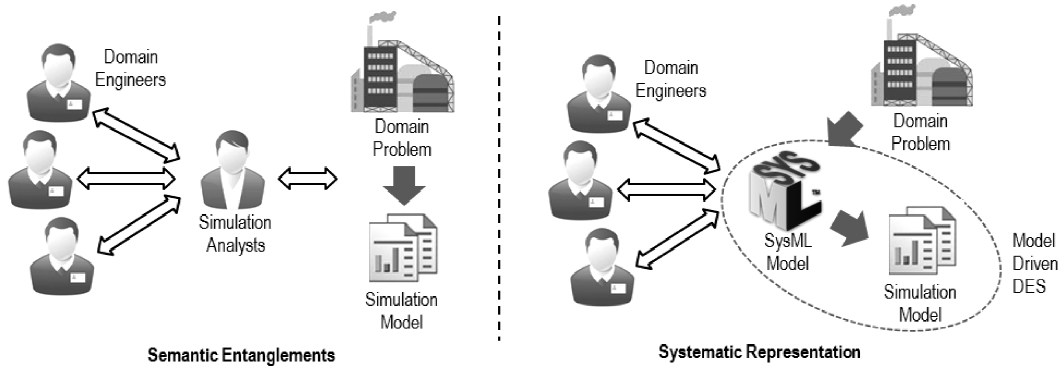


Figure 9.1 Frameworks of traditional modeling process and model driven approach

Recently, SysML has emerged as the preferred method for modeling complex systems. It has also been considered for use in the conceptual design phase in a systematic product design process. It is proposed to describe the system more clearly and reduce the effort for any further modification to the system model. It can not only be regarded as a common language between domain engineers, but also a source code which can be automatically transformed into Arena simulation language. While it is a formal language, conforming to Meta-Object Facility, it has a graphical user interface, making most diagrams relatively easy to understand. The block is the basic unit of presentation in SysML and can be used to represent hardware, software, facilities, personnel, or any other system element. Typically, there are four kinds of SysML diagrams, i.e., structure diagram, behavior diagram, requirements diagram, and parametric diagram as shown in Figure 9.2.

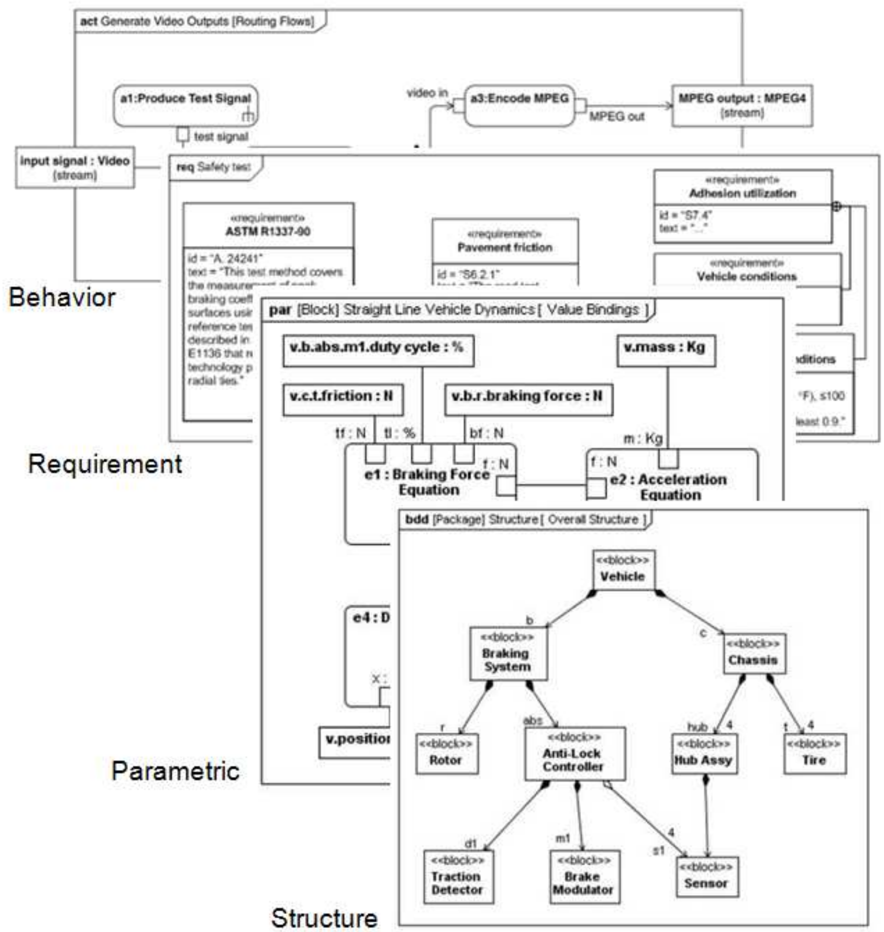


Figure 9.2 The four pillars of SysML (Friedenthal et al., 2014)

9.3 DES simulation model

In Arena, modules are basic elements that represent processes or logic. Connector lines are used to join these modules together and specify the flow of entities. While modules have specific actions relative to entities, flow, and timing, the precise representation of each module and entity relative to real-life objects is subject to the analysts. Arena is a process-oriented modeling tool for discrete-event systems. In other words, the modeling in Arena environment is structured as a workflow of stepwise activities and actions.

Other than building the simulation models by Graphical User Interface (GUI) operations, Arena can be integrated with Microsoft technologies, including Visual Basic, Access database and ActiveX controls. Through such interface, DES modeling transformation and automation become possible. Arena simulation model is run in sequence of events, and the events can be standard modules of Arena or Visual Basic for Applications (VBA) custom codes. A basic sequence of events in an Arena model is shown in Figure 9.3.

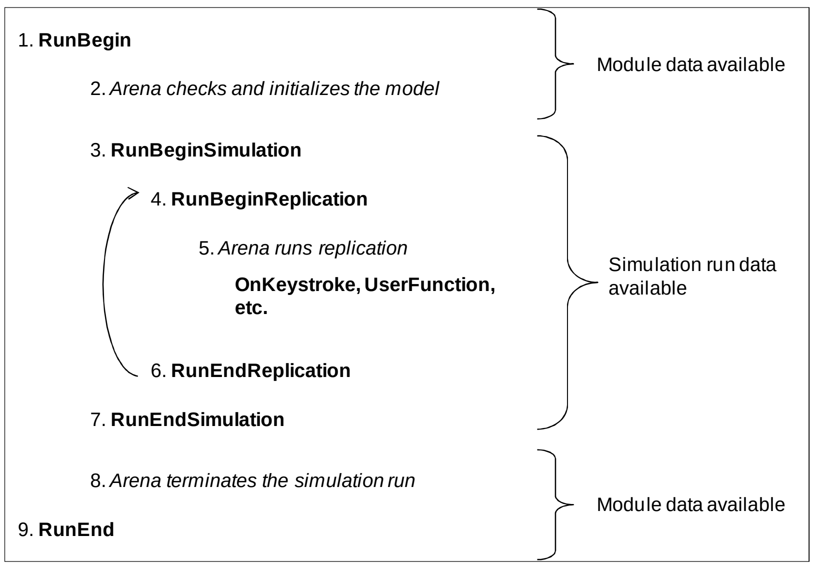


Figure 9.3 Arena/VBA sequence of events

The basic building blocks are elements, such as entities, queues, resources and sequences, and the process blocks which affect them, some of which are shown in Figure 9.4.

Create 1	Push (possibly) batches of entities into the model with a (possibly) random time between.	Assign 1	Assign values (especially Attributes) when an entity passes through.
Process 1	Models Queue-Seize-Delay-Release of Resource, or any part of this (like pure Delay).	Record 1	Record information when entities pass through, typically statistics on entities.
Decide 1	Make decisions about where to go next based on conditions or chance.	Batch 1	Combine multiple entities into a single entity.
Dispose 1	Take entities out of the model and (perhaps) record statistics.	Separate 1	Split multiple entities that were combined, or duplicate a single entity.

Figure 9.4 Primitive modules in Arena simulation model

The main task of model transformation is to set the rules in transformation codes to convert SysML activities into Arena elements and modules.

The other important function of Arena is the data input analyzer. A valid DES model will not work properly without a valid input data. Normally the data collected from real world are limited, input data fitting is then necessary. Such data is commonly governed by certain kinds of distributions and input analyzer is used to pre-process the raw input data to find a marching distribution property. This data pre-process procedure also should be automated and become one part of the SysML driven DES process. However, in modern assembly systems, a online real-time data collection become available. With real-time collected data sets, the Arena model can also run based on real system data without any regression and distribution fitting.

9.4 Real-time Data Collection

Machines in factory floor today are much more connected than in the past, with more networking options available through the rising popularity of Ethernet, open-architecture machine communications protocol is becoming more available. For companies looking to improve their performance, getting data off the factory floor in real-time is critical. Especially in assembly systems, as the last step of product variety realization, the various demands and input parts variance always is a dynamically changing and affects the systems performance. A real-time collected data with the help of DES simulation tools would assist immediate decision making on process plans to keep system performance level.

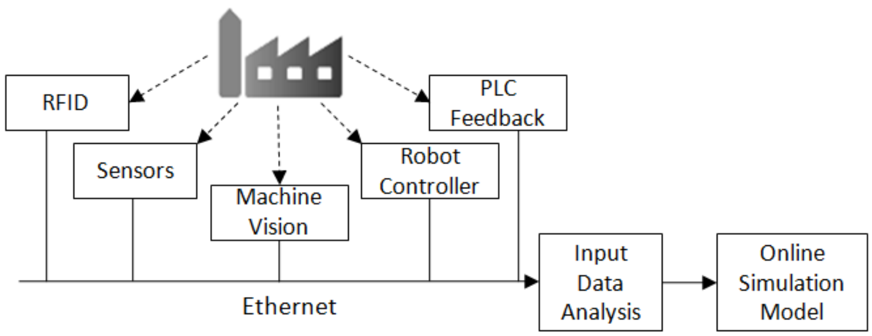


Figure 9.5 Shop floor real-time data collection infrastructure

As shown in Figure 9.5, by using different technology, all kinds of production data can be real-time collected and feed into online simulation model. Product information can be collected via Radio-frequency Identification (RFID), which allows identification of part numbers under the high mix production environment. Different kinds of sensors and machine vision can collect data of material flow in the whole assembly system. Robot controller and Programmable Logic Controller (PLC) as the actuation part of the system, would able to provide production condition data such as cycle time, the process time in each assembly motion.

With all the data collected, an input data analysis is needed to convert raw data to useful production status information. Input data analysis would convert the raw data into meaningful information to correct the simulation model. Such analysis can be automatically executed and evaluated based on Log Likelihood, Chi-square goodness-of-fit test, and K-S goodness-of-fit test. The collected data can be used to correct the DES simulation parameters.

9.5 Online Simulation-based Planning of Assembly Process

The model-driven DES simulation model in Arena can be directly translated from a SysML model of the assembly system. Based on the system layout design from Chapter 8, by adding running parameters and material flows in between different departments, a high level SysML model can be derived. In order to ensure a relativity high accurate representation of the physical system, the detailed work stations and key components such as robots, feeders, and assembly actuators should be presented with more details. The modeling granularity depends on the output requirements and system complexity. High level outputs such as overall through put or Overall Equipment Effectiveness (OEE) can reach relatively high accuracy without too much modeling details. On the other hand, some detailed outputs in smaller scale of system, such as a utilization rate of certain type of machines, needs finer modeling details. A large scale simulation model with complex details could also lead to computational difficulty especially in online simulation.

The model transformation process allows almost any personal in factory floor without domain knowledge of DES simulation to build the assembly system's logic in SysML using Arena semantics. The input files of model transformation are XMI files of SysML models and text based data files. The XMI files contents SysML action diagrams which have all modeling information of Arena events to represent assembly systems. As the common composition of an arena event is known, I can build an Arena processes and elements template which stores all the information needed for a standard assembly system. Then the workload of building SysML model will be reduced as most of the standard processes and elements can be directly picked from template and no further definition is needed. Data input analyzer is a plug-in of Arena and its data fitting function relies on user manual inputs. A third party data fitting program using DFitTool of Matlab is used in order to gain the automatic data conversion and distribution parameter generation. The overall model translation process framework is shown in Figure 9.6.

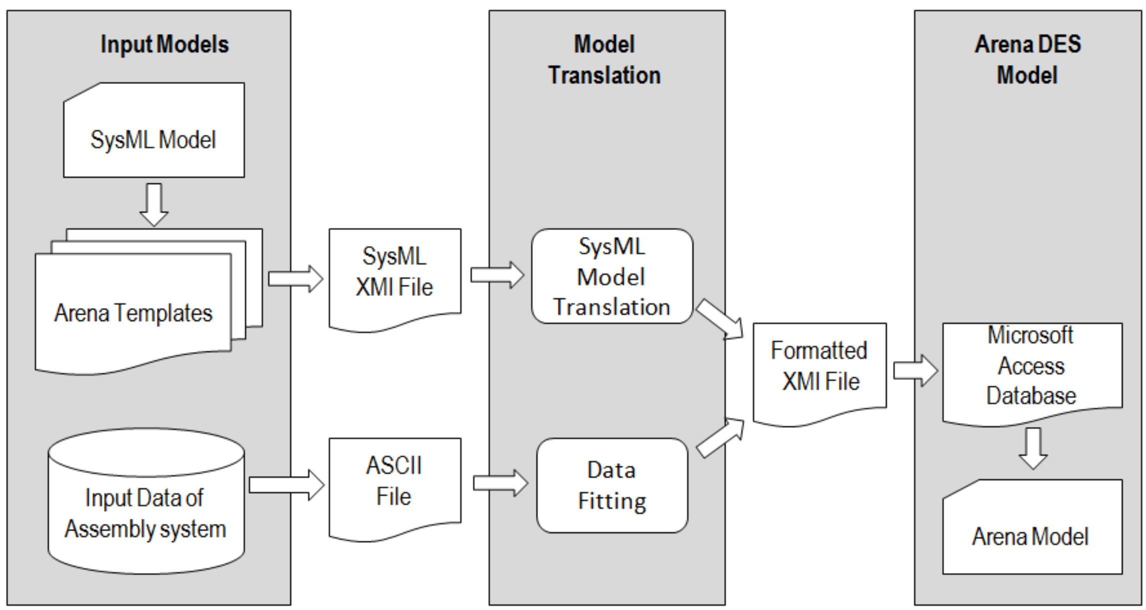


Figure 9.6 Model translation framework

The XMI files in the translation process will be converted several times in a Java environment. Input data including the assembly system layout design and machine running parameters will also be formatted into a proper format that enables reading through the Matlab package. The data will be fitted and a parameter set will be generated and saved to the formatted XMI files. MDB files of Microsoft access serves as an intermediate to the XMI file and Arena DOE format model. All the translation can be packed into one executable program and operated.

A properly built Arena simulation can run much faster than real life process, which makes the simulation based optimization possible. Through program controlled simulation iterations, the assembly systems performance can be evaluated in different conditions. Resource allocation schemes and process plan under different product variety demand can be optimized by using the simulation results. A real-time feedback to the factory floor is possible if all the production parameter is gathered in real-time. By using the production data, the simulation model can be continuously refined and provide decision making support to the assembly planning process. By using the experiment tool, a what-if analysis can also be developed to simulate the system response to the varying demands and configuration change, as shown in Figure 9.7.

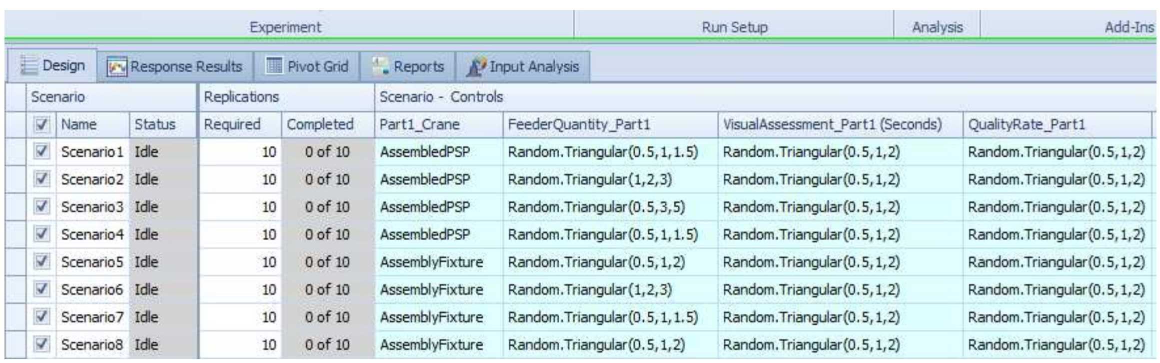


Figure 9.7 Experiment tool in Arena for what-if analysis

Figure 9.8 shows the framework of the online simulation with real-time data feedback. The initial input data will be used to build the SysML model of the assembly system, including system layout, production demands, process sets and their routing in the system. Then the model will be translated automatically into an Arena DES model. Through the simulation and what-if analysis, the analysis result will be used to refine the actual system operation parameters as well as modification to the simulation model itself. When the output of the physical system changed, either because of the operation parameter or environment constraint change, new set of data will be feedback to system model which drives the simulation model.

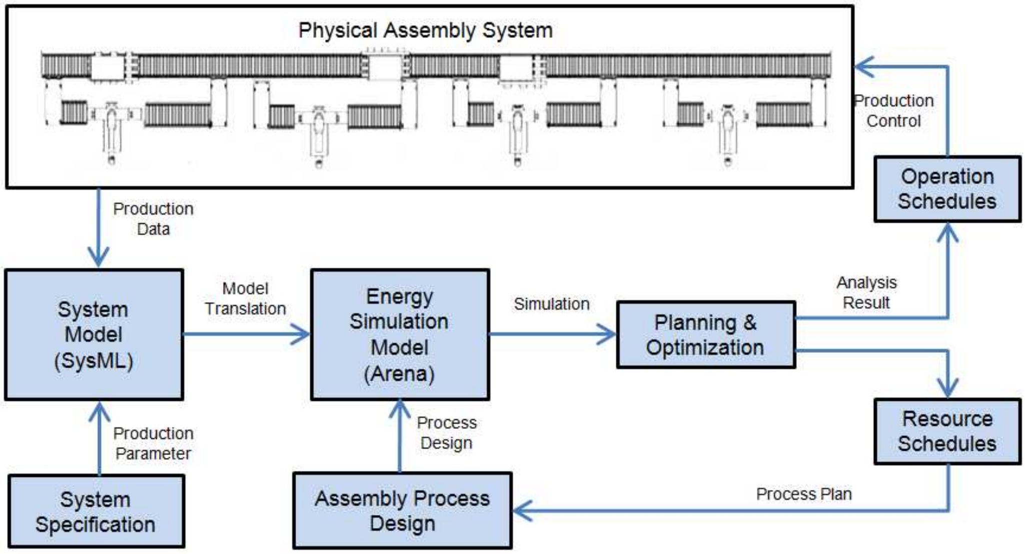


Figure 9.8 Online simulation and planning framework

In the game theoretic optimization of assembly system design, it is also possible to use the simulation result as fitness function of the LFGA search algorithm when a more accurate result is needed. However, this approach is currently limited by the computational power of personal computer, which would be very slow even when dealing with very small scale systems.

9.6 Discussions

In high variety assembly system design problems, the most complicated part is to handle variety in assembly process planning. High product variety always leads to highly mixed production schedule and the process planning complexity will grow exponentially with the incensement of mixed product number in system. With the game theoretic optimization solution provided through Chapter 4 to 8, the problem can be solved based on a LFGA search under the proven evaluation criteria. However, a simulation based solution is also necessary, as not only it can be used as a verification of the assembly system design, but also it will serve as a filed application to help the running production and improvement after the system has been deployed.

The data driven simulation tool provides powerful calculation capability to provide real-time feedback to the running assembly system, and the online data input brings even better reliability on the simulation results. I also explored the possibility of simulation-based game theoretic optimization by using the SysML based parameter driven simulation as fitness function to the LFGA search. However, the computational load is much higher than the mathematical LFGA thus hardly reaches convergence in a reasonable time frame.

9.7 Summary

The original achievements in this chapter can be concluded as followings:

1) The model driven architecture can translate a generic SysML model of the assembly system into a Arena simulation model, and the GPPS data are also used as input to parameterize the simulation model;
2) The online data acquisition and real-time simulation based decision making feedback architecture can handle the highly dynamic high variety assembly process planning problem.

In summary, this chapter provides an online data driven simulation model to assist high variety assembly process planning on factory floor with utilization of available factory production data. It provides a verification method to the assembly system design as well as explored the potential of simulation-based optimization in high verity assembly system design.

CHAPTER 10

HIGH VARIETY ASSEMBLY SYSTEM DESIGN CASE STUDY

A case study of high variety automobile connector assembly system design is used to illustrate the proposed design and optimization method. In this chapter, by developing the assembly system from product variety demands to process design and planning, resource allocation, layout design and simulation, it validates the game theoretical optimization method on solving high variety assembly system design problems.

10.1 Introduction

Automotive industry is a huge and still fast growing industry, which is ranked as one of the world's most important economic sectors by revenue. Being an industry with over a century's history, it is still growing fast, not only in production volume, but also in product variety. A modern car contains several hundreds of connector parts and most of them are different. Due to the complexity of modern car's electrical system, the automotive connectors design is very complex with many variants to meet different requirements and standards. Especially for the critical connectors on the control system and safety systems, the sealed interlock connectors used in these subsystems have very high cost in manufacturing. The major reason is not the complexity of the design but the product's high variety and volatile demand.

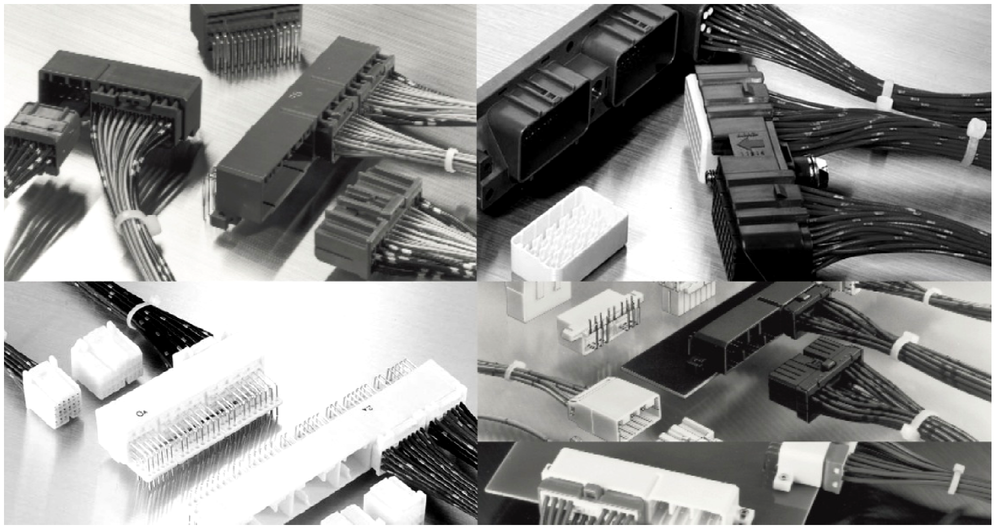


Figure 10.1 Some automotive interlock connector families

The interlock connector product family has more than five hundred different varieties and the annual demand for them varies from tens to hundred thousands. So it is very expensive to develop delicate assembly system for each of the products. The traditional solution is to build delicate assembly system for just several major product variants and manually assemble the rest. However, the market trends has changed and demand of these products become much more fickle, with increasing labor cost. It is also harder to justify the cost of delicate assembly systems as each product variant's life cycle is reducing while their variety keeps increasing. As a result, the high demand to provide an assembly system design to accommodate high variety automotive car interlock connectors leads to this case study.

10.2 Process Variety Derivation

10.2.1 Product Information

The product family selected for this case study is a seven-part assembly with 4 to 30 position interlock connector. The product varieties include number of pin positions, colors, latch opinions, sealing options, wire options, and bending options. There are totally 79 part numbers involved and have 864 product variants. Table 10.1 shows the part list for this product family.

Table 10.1 Part list of the interlock connector family

Part ID	**Part Type**	**Part Level(L^p)**	**Color**	**Part ID**	**Part Type**	**Part Level(L^p)**	**Color**
H01-4-1	Housing - Outer	3	Black	H02-20-1	Housing - Inner	3	Black
H01-4-2	Housing - Outer	3	White	H02-20-2	Housing - Inner	3	White
H01-4-3	Housing - Outer	3	Yellow	H02-20-3	Housing - Inner	3	Yellow
H01-6-1	Housing - Outer	3	Black	H02-24-1	Housing - Inner	3	Black
H01-6-2	Housing - Outer	3	White	H02-24-2	Housing - Inner	3	White
H01-6-3	Housing - Outer	3	Yellow	H02-24-3	Housing - Inner	3	Yellow
H01-8-1	Housing - Outer	3	Black	H02-30-1	Housing - Inner	3	Black
H01-8-2	Housing - Outer	3	White	H02-30-2	Housing - Inner	3	White
H01-8-3	Housing - Outer	3	Yellow	H02-30-3	Housing - Inner	3	Yellow
H01-12-1	Housing - Outer	3	Black	I01-1	Insert - Short	2	Red
H01-12-2	Housing - Outer	3	White	I02-1	Insert - Mid	2	Red
H01-12-3	Housing - Outer	3	Yellow	I03-1	Insert - Long	2	Red
H01-16-1	Housing - Outer	3	Black	L01-4-1	Latch	3	Red
H01-16-2	Housing - Outer	3	White	L01-6-1	Latch	3	Red
H01-16-3	Housing - Outer	3	Yellow	L01-8-1	Latch	3	Red
H01-20-1	Housing - Outer	3	Black	L01-12-1	Latch	3	Red
H01-20-2	Housing - Outer	3	White	L01-16-1	Latch	3	Red
H01-20-3	Housing - Outer	3	Yellow	L01-20-1	Latch	3	Red
H01-24-1	Housing - Outer	3	Black	L01-24-1	Latch	3	Red
H01-24-2	Housing - Outer	3	White	L01-30-1	Latch	3	Red
H01-24-3	Housing - Outer	3	Yellow	S01-4-1	Seal	1	Black
H01-30-1	Housing - Outer	3	Black	S01-4-2	Seal	1	Blue
H01-30-2	Housing - Outer	3	White	S01-6-1	Seal	1	Black
H01-30-3	Housing - Outer	3	Yellow	S01-6-2	Seal	1	Blue
H02-4-1	Housing - Inner	3	Black	S01-8-1	Seal	1	Black
H02-4-2	Housing - Inner	3	White	S01-8-2	Seal	1	Blue
H02-4-3	Housing - Inner	3	Yellow	S01-12-1	Seal	1	Black
H02-6-1	Housing - Inner	3	Black	S01-12-2	Seal	1	Blue
H02-6-2	Housing - Inner	3	White	S01-16-1	Seal	1	Black
H02-6-3	Housing - Inner	3	Yellow	S01-16-2	Seal	1	Blue
H02-8-1	Housing - Inner	3	Black	S01-20-1	Seal	1	Black
H02-8-2	Housing - Inner	3	White	S01-20-2	Seal	1	Blue
H02-8-3	Housing - Inner	3	Yellow	S01-24-1	Seal	1	Black
H02-12-1	Housing - Inner	3	Black	S01-24-2	Seal	1	Blue
H02-12-2	Housing - Inner	3	White	S01-30-1	Seal	1	Black
H02-12-3	Housing - Inner	3	Yellow	S01-30-2	Seal	1	Blue
H02-16-1	Housing - Inner	3	Black	P01-1	Pin	1	Gold
H02-16-2	Housing - Inner	3	White	W01-1	Wire - Short	1	Black
H02-16-3	Housing - Inner	3	Yellow	W02-1	Wire - Long	1	Black

Each connector contains one outer housing, one inner housing, one side insert to lock the housings, one optional latch, one optional sealing ring, pins and optional wires.

The variety options are shown in Table 10.2. A total number of 864 product varieties are introduced by these options.

Table 10.2 Variety options of the interlock connector family

Varity Option	Choice of Varity	Variety Introduced
Number of Pin Positions	4, 6, 8, 12, 16, 20, 24, 30	8
Connector Color	Black, White, Yellow	3
Latch Option	With latch, Without latch	2
Sealing Option	Silicone seal, NBR seal, No seal	3
Pin Bending	Bending, No bending	2
Wire Option	No wire, Short wire, Long wire	3

Table 10.3 Available machine list

Machine ID	Machine Type	Priority Level(L^m)	Availability (hour)	Machine ID	Machine Type	Priority Level(L^m)	Availability (hour)
M01	Inline Molding	2	250	M20	Wire Braiding / Trimming	1	50
M02	Laser Cutting	3	200	M21	Strip Cutter	1	50
M03	Ultra-Sonic Cleaning	1	250	M22	Strip Cutter	1	50
M04	Ultra-Sonic Cleaning	1	250	M23	Robotic Kitting	3	100
M05	Flexible Feeding	3	50	M24	Robotic Kitting	3	100
M06	Flexible Feeding	3	50	M25	Robotic Insertion	3	50
M07	Bowl Feeder	1	100	M26	Robotic Insertion	3	50
M08	Bowl Feeder	1	100	M27	Robotic Insertion	3	50
M09	Strip Feeder	1	250	M28	Robotic Insertion	3	50
M10	Strip Feeder	1	250	M29	Automatic Insertion	2	60
M11	Stitching	2	100	M30	Automatic Insertion	2	60
M12	Stitching	2	100	M31	Automatic Glue Dispenser	1	150
M13	Variable Pitch Stitch/Bend	3	100	M32	Laser Mark Printer	3	250
M14	Pin Bending	1	150	M33	Vision Inspection	3	40
M15	Pre-tinning	1	200	M34	Vision Inspection	3	40
M16	Pre-tinning / Soldering	3	100	M35	Vision Inspection	3	80
M17	Laser Soldering	3	150	M36	Ultra-Sonic Inspection	2	120
M18	Automatic Soldering	2	150	M37	Ultra-Sonic Inspection	2	120
M19	Automatic Soldering	2	150	M38	Pull Tester	1	250

All aspects of process design and resources are coded individually, including processing methods and their priority levels, along with the corresponding machine tools and their identifications. Each type of machine tools may have multiple instance as the backup, each of which need to be uniquely identified. Table 10.3 lists all the available assembly resources with their priority level and availability. Some machines have shorter available time is due to the high demand of such machines inside the plant. In Table 10.4, the list of assembly process is presented with the related part ID, machine ID, average setup time, and average processing time.

Table 10.4 Assembly process list

Process ID	**Process Type**	**Related Part ID(T^{SET} (s))**	**Machine ID (T^{p} (s))**
P01	Over-molding	H01-x-x(5)	M01(2)
P02	Braiding	W01-x(3), W02-x(5)	M20(10)
P03	Wire Cutting	W01-x(3), W02-x(5)	M02(2), M20(1), M21(0.5), M22(0.5)
P04	Wire Trimming	W01-x(3), W02-x(5)	M20(2)
P05	Cleaning	H01-x-x(5), H02-x-x(5)	M03(3), M04(3)
P06	Pre-tinning	W01-x(3), W02-x(5), P01-1(1)	M15(2), M16(3)
P07	Wire Feeding	W01-x(3), W02-x(5)	M05(3), M06(3), M09(0.5), M10(0.5)
P08	Soldering	W01-x(3), W02-x(5), P01-1(1)	M16(3), M17(0.5), M18(1), M19(1)
P09	Pin Cutting	P01-1(1)	M02(2), M21(0.5), M22(0.5)
P10	Pin Feeding	P01-1(1)	M05(3), M06(3), M07(1), M08(1), M09(0.5), M10(0.5)
P11	Stitching	P01-1(1), H02-x-x(5)	M11(0.5), M12(0.5), M13(1)
P12	Bending	P01-1(1)	M13(0.5), M14(0.2)
P13	Feeding	H01-x-x(5), H02-x-x(5), I0x-1(3), L01-x-x(5), S01-x-x(3)	M05(3), M06(3), M07(1), M08(1)
P14	Seal Insert	H02-x-x(5), S01-x-x(3)	M25-28(2), M29(1),M30(1)
P15	Housing Insert	H01-x-x(5), H02-x-x(5)	M25-28(2), M29(1),M30(1)
P16	Key Insert	H01-x-x(5), H02-x-x(5), I0x-1(3)	M25-28(2)
P17	Latch Insert	H01-x-x(5), L01-x-x(5)	M25-28(3)
P18	Gluing	H01-x-x(5), H02-x-x(5), I0x-1(3)	M31(3)
P19	Pull Test	W01-x(3), W02-x(5), P01-1(1), H01-x-x(5)	M38(1)
P20	Mark Printing	H01-x-x(5)	M32(0.5)
P21	Packaging	H01-x-x(5)	M23(5), M24(5)
P22	Solder Inspection	W01-x(3), W02-x(5), P01-1(1)	M36(1), M37(1)
P23	Surface Inspection	H01-x-x(5), H02-x-x(5), I0x-1(3), L01-x-x(5)	M33-35(2)
P24	Position Inspection	H01-x-x(5), H02-x-x(5), I0x-1(3), L01-x-x(5), S01-x-x(3)	M33-35(2)

10.2.2 Product GPPS

With all the product and process information given above, GPPS for this product family can be constructed. As different assembly process design would lead to different GPPS, there will be as many GPPS as the total number of available design sets, one of the GPPS is shown in Figure 10.2. As the component $\{H01\}$, $\{H02\}$ and $\{L01\}$ have more than 8 instances, they are not shown in this GPPS to save space.

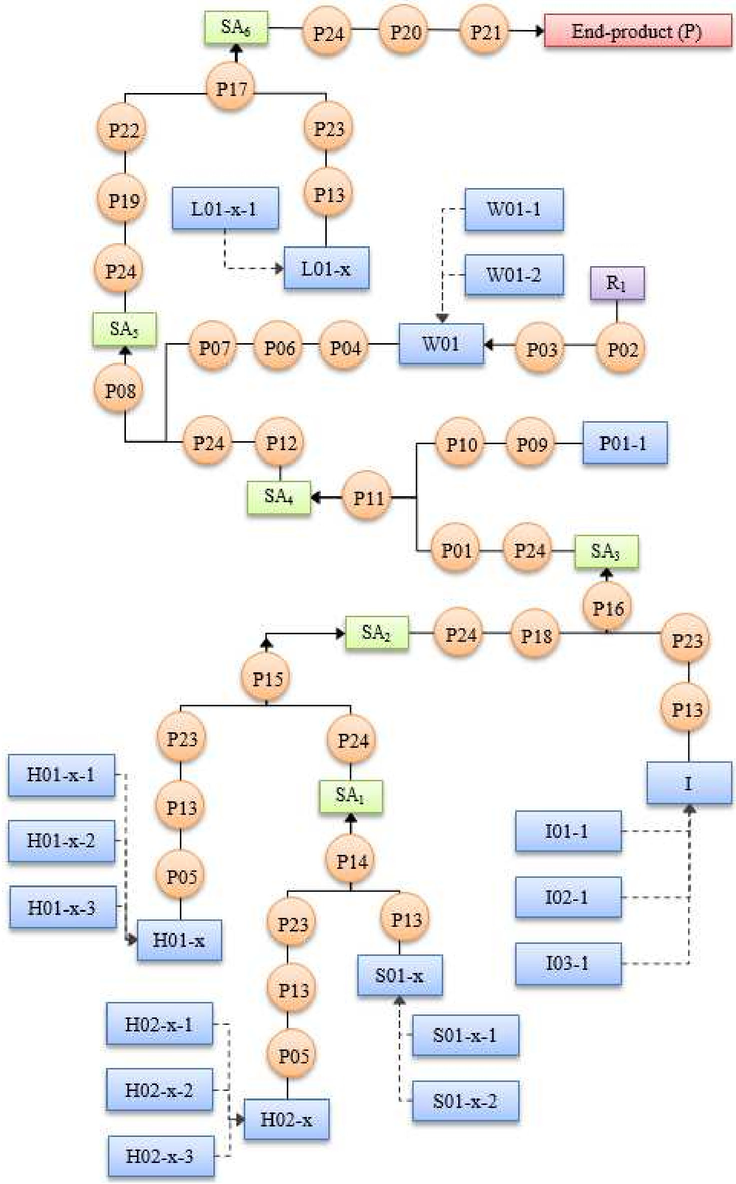


Figure 10.2 GPPS of automotive interlock connector families

R_1 is the wire roll before cutting and braiding, so considered as raw material in the system. By choosing different processes and components in this GPPS, the product varieties can be explored using Matlab code. By using rule mining technique, a full range of possible assembly process sets are mapped form given product variety and their demands. A resource allocation set is also produced in this step to ensure all resource allocation plans are feasible based on the given relationship of assembly processes and their possible resource usage.

10.3 Game Theoretic Optimization of Assembly Process Design and Planning

After having collected all the data for the given product family, the key part of the high variety assembly system design, assembly process design and process planning problem, can be solved using game theoretic optimization solution.

10.3.1 Game Theoretic Optimization Formulation

In the high variety connector assembly system, the process design has a significant impact on the system performance. As the volatile demand will not allow such assembly system to be designed and built as a traditional dedicate assembly line, the process design must be carefully tailored to achieve high flexibility. However, the emphasize on flexibility would have negative impact on the factory load balancing. As when the flexibility become the only goal of the system design, some of the more integrated and flexible machine get too much load while some dedicate machines are totally forgotten. And in fact, a good arrangement of the given dedicate machines based on the character of the product family sometimes outperforms the all-in-one flexible machines. So in the phase of assembly resource allocation, to maximize the utilization of all available machines and leverage the utilization of older dedicate machines and new flexible machines is also critical. The competing goal in process design and resource allocation can be formulated as a game theoretic optimization.

The process design will take place first so it is formulated as leader. Use Eq. (7.1) as the design variable in the upper level function, based on the assembly process plan set, the setup time can be further determined using T^{SET} data in Table 10.4. The resource allocation plan will be decided after the process design, it is formulated as follower. Use Eq. (7.2) as the design variable in the lower level function, the process time of each assembly process then can be determined based on the resource allocation plan and T^{p} data in Table 10.4.

10.3.2 Mathematical Model

Based on the above formulations, the game theoretic optimization model can be used for the connector product family assembly process design and resource allocation, as shown in Eq. (10.1). The model consists of the upper- and lower-level optimization problems. The upper level optimization aims at maximization of process commonality (*CI*), whilst the lower level is geared towards maximization of resource utilization (*E*). The upper-level objective function is composed by considering process flexibility index *C*, sequencing flexibility index *F*, and process feasibility index *A*.

To maximize process commonality at the upper level, first a process design scheme x is determined, which will affects total setup time T^{SET}. Based on the upper level's decision, the lower level seeks for a resource allocation plan y that achieves a maximal utilization rate. Difference in resource allocation plans affects total process time

T^p at workstations and that of the parts $\sum_{j=1}^{u} t_{ij}^{p}$. It further feeds back to the upper-level problem and influences the calculation result of process flexibility index *C* and sequencing flexibility index *F*, and ultimately regulates the achievement of process commonality index *CI*. Then at the upper level, feedback of resource allocation *y* from the lower level takes part in the process design decision making regarding *x* in order to maximize process commonality. Such *x* decisions are further passed on to the lower level, which is guided by *x* as a parametric optimization problem. This process iterates until both the upper and lower levels arrive at their optima, when the optimization reaches to convergence.

Throughout the optimization process, the upper-level takes a leader's role in such a way that the process commonality index is optimized first with its objective function valuates according to both process design scheme *x* and resource allocation plan *y*. While optimal process design decision making regarding x considers feedback from the lower-level resource allocation decisions *y*, the lower-level problem acts as a follower to make decision making of *y* conform to the upper-level *x* decisions. Finally the model returns an equilibrium solution that leverages both the upper- and lower-level optimization problems, which represents equilibrium between the assembly process design in terms of process commonality, flexibility and efficiency.

$$Max\,CI(x,y) = Max\left(\frac{C(x,y)A(x)}{F(x,y)}\right)$$

$$s.t. \qquad x \in X$$

$$C(x,y) = \frac{\sum_{p=1}^{n_p}\sum_{i=1}^{n_d} \lambda_{ixp} T^{SET} T^{p}(y) \sum_{i=1}^{n_d} D_i}{\sum_{p=1}^{n_p} \eta_p \sum_{i=1}^{n_d}\left(D_i \sum_{j=1}^{u} (t_{ij}^{SET} \cdot t_{ij}^{p}) \right)}$$

$$A(x) = 1 - \frac{\sum_{i=1}^{n_d}\sum_{j=1}^{u} K\left(L_i^p - L_{ij}^m\right)^2}{\sum_{i=1}^{n_d}(L_i^p)^2 + \sum_{i=1}^{n_d}\sum_{j=1}^{u}(L_{ij}^m)^2}$$

$$K = \begin{cases} 0.01 & if\ L_i^p > L_{ij}^m \\ 0.02 & if\ L_i^p \le L_{ij}^m \end{cases}$$

$$F(x,y) = 1 + \sum_{e=1}^{n_e}\left(\frac{\sum_{h=1}^{u'}\sum_{k=1}^{u'}\left((\xi_{ye})_{hk}(\alpha_{ye})_{hk} - \frac{\sum_{i=1}^{u'}\sum_{j=1}^{u'}((\xi_{ye})_{ij}(\alpha_{ye})_{ij})}{\sum_{i=1}^{u'}\sum_{j=1}^{u'}(\alpha_{ye})_{ij}} \right)^2}{\sum_{i=1}^{u'}\sum_{j=1}^{u'}(\alpha_{ye})_{ij}} \right)$$

$$\eta_p = \begin{cases} 1 & if\ \forall x_{iaj} \in x,\ x_{iaj} - p = 0, i \in \{1,2,\cdots,n_d\}, a \in \Omega_i, j \in \{1,2,\cdots,u\} \\ 0 & if\ \forall x_{iaj} \in x,\ x_{iaj} - p \ne 0, i \in \{1,2,\cdots,n_d\}, a \in \Omega_i, j \in \{1,2,\cdots,u\} \end{cases}$$

$$\lambda_{ixp} = \begin{cases} 1 & if\ p \in x \\ 0 & if\ p \notin x \end{cases}$$

$$t_{ij} > 0$$

$$(\xi_{ye})_{hk} > 0,\ (\xi_{ye})_{ij} > 0$$

$$(\alpha_{ye})_{hk} = \begin{cases} 1 & if\ (\xi_{ye})_{hk} = 0 \text{ or } (\xi_{ye})_{hk} = \infty \\ 0 & if\ (\xi_{ye})_{hk} \ne 0 \text{ and } (\xi_{ye})_{hk} \ne \infty \end{cases}$$

$$(\alpha_{ye})_{ij} = \begin{cases} 1 & if\ (\xi_{ye})_{ij} = 0 \text{ or } (\xi_{ye})_{ij} = \infty \\ 0 & if\ (\xi_{ye})_{ij} \ne 0 \text{ and } (\xi_{ye})_{ij} \ne \infty \end{cases}$$

$$
\begin{aligned}
&T^{SET} = \sum_{e=1}^{n_e} T_e^{SET} \\
&y \in \arg\ Max\ E(x,y) = Max\left(\left(1-\exp\left(-\psi\left(\frac{\sum_{e=1}^{n_e} T_e^{p}}{\sum_{e=1}^{n_e} T_e^{SET}}\right)\right)\right)\left(\frac{\sum_{e=1}^{n_e} T_e^{p} + \sum_{e=1}^{n_e} T_e^{SET}}{\sum_{e=1}^{n_e} T_e^{a}}\right)\right) \\
&s.t. \quad y \in Y \\
&\qquad T_e^{p} < T_e^{a} \\
&\qquad T_e^{SET} = \sum_{h=1}^{u'}\sum_{k=1}^{u'} (\beta_{ye})_{hk}^{z^*}\ (\xi_{ye})_{hk} + S_{FL} \\
&\qquad \sum_{h=1}^{u'}\sum_{k=1}^{u'} (\beta_{ye})_{hk}^{z^*}\ (\xi_{ye})_{hk} \le \sum_{h=1}^{u'}\sum_{k=1}^{u'} (\beta_{ye})_{hk}^{z}\ (\xi_{ye})_{hk} \left(z=1,2,\cdots,n_z, e=1,2,\cdots,n_e\right)
\end{aligned}
\tag{10.1}
$$

10.3.3 Model Solution

Figure 10.3 illustrates the solution process of game theoretic optimization, which proceeds as the following:

Step1: Generate process design schemes for all the parts and represent them as $\mathbf{X}$ vector set;

Step2: Select a process scheme *x* from $\mathbf{X}$;

Step3: Generate resource allocation plans for each process design scheme *x*, forming a resource allocation plan set $\mathbf{Y}$;

Step4: Generate a number of n_e setup time matrixes $\mathbf{H}_{ye}$, along with a number of n_z possible processing sequence plans for all resource allocation plans $y \in \mathbf{Y}$;

Step5: Find a processing sequence plan z^* with the shortest setup time based on $\mathbf{H}_{ye}$ by maximizing resource utilization E;

Step6.1: Calculate T^{SET} and E;

Step 6.2: Generate a setup time table for each part based on its setup time matrix and the corresponding optimal sequence plan z^*, Step 6.3: Derive the optimal resource allocation plan y^* corresponding to sequence z^*;

Step 6.4: Calculate F value for y^*;

Step 7: Calculate *C*, *A*, and *CI*;

Step 8: Record values of *CI* and *E*, along with the corresponding values of x, y, and z;

Step 9: Repeat Steps 2-8 until upper-level optimum *CI* is returned, along with the lower-level optimum *E*.

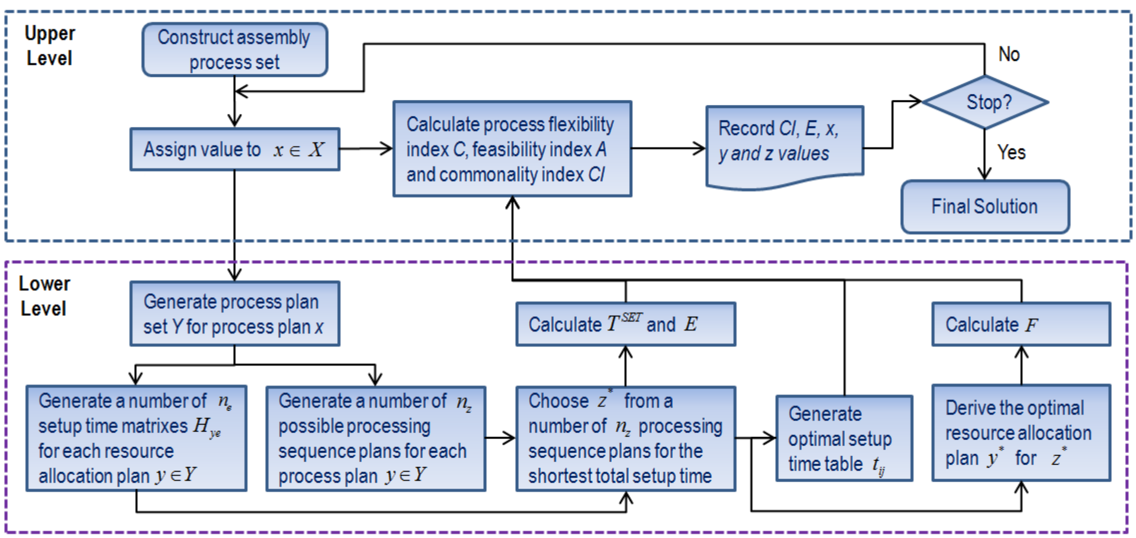


Figure 10.3 Solution framework of the connector assembly system design optimization

10.3.4 LFGA Solution

Based on the LFGA solution explained in Chapter 5, the upper-level process design can be represented as chromosome with length of n_d, in which each gene indicates a corresponding processing method ID for the respective part. For each process design *x*, there are a number of *v* resource allocation plan sets that can be generated. Each resource allocation plan *y* is represented using another chromosome with length of $u \times n_d$. Within this resource allocation chromosome, starting from the first gene, every segment of a number of *u* genes corresponds to a resource allocation plan, while each gene represents the index of a resource cluster. Throughout the LFGA reproduction process, a gene of a resource cluster performs as the basic operational unit for crossover and mutation of the chromosomes. Figure 10.4 illustrates the LFGA encoding scheme for decision variables *x* and *y* of product design and planning.

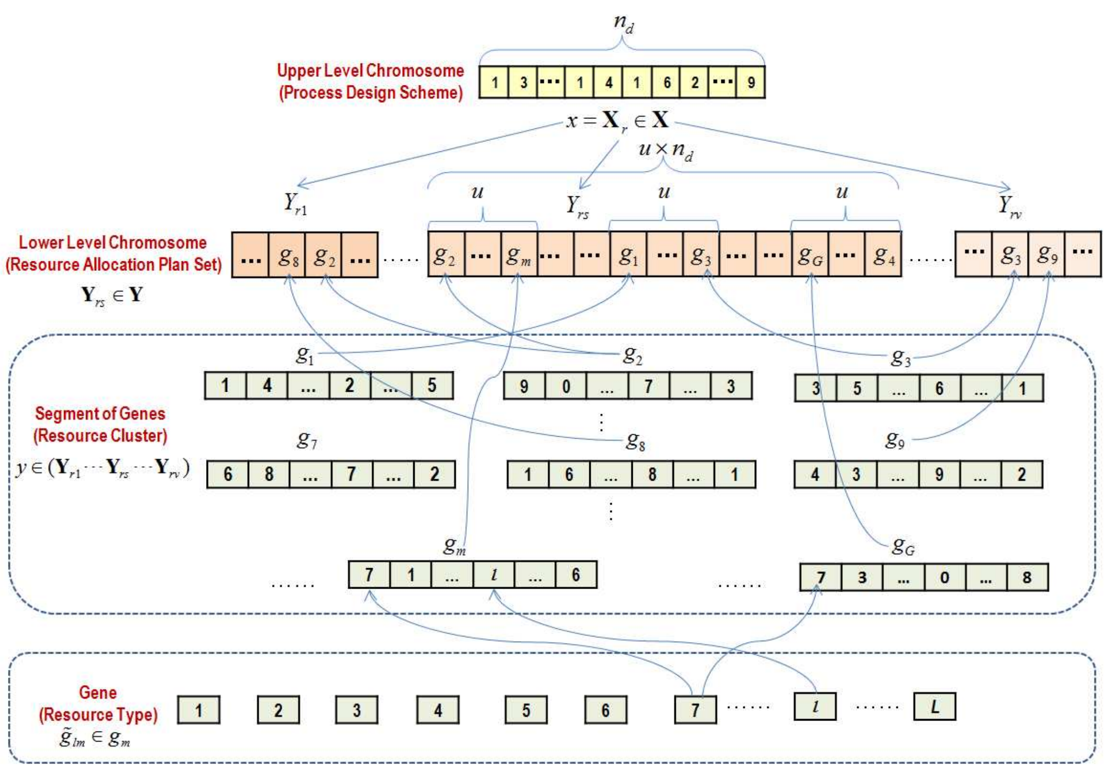
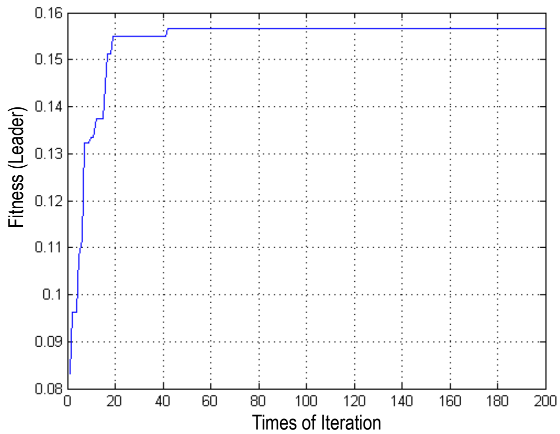


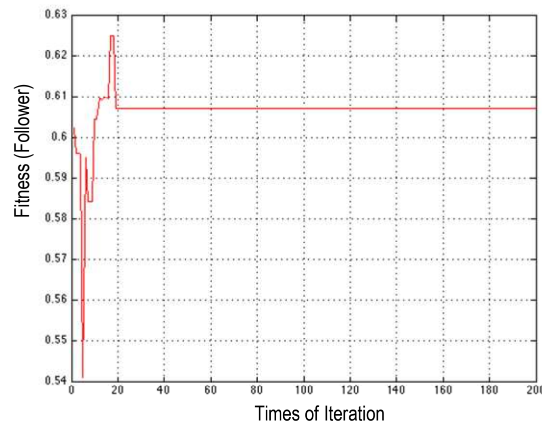
Figure 10.4 LFGA encoding of decision variables *x* and *y*

To improve computational efficiency, a number of *G* resource clusters can be generated based on a priori knowledge about the domain problem, which is the resource list and resource to process relationship data in Table 10.3 and Table 10.4. Each resource cluster is represented as a segment of genes. The maximal length of a segment is $L \in N^+$, which is the total number of resource types available in a shop floor with each gene representing a specific type of resources. It is important for mutation and crossover operators to handle the resource constraints defined in the process sets and generate off springs that are technically feasible for process planning. The fitness function is supposed to perform as a screening criterion for selecting the appropriate upper- and lower-level chromosomes. In the leader-follower model, I adopt the upper- and lower-level objective functions in Eq. (10.1) to be the respective fitness functions of process design x and resource allocation y.

In GA implementation, we set the population size to1000 and use a mutation rate of 0.05 and a crossover rate of 0.8. After 200 iterations, the upper-level fitness value converges at 0.1566, as shown in Figure 10.5(a), whereas the lower-level fitness value converges at 0.6072, as shown in Figure 10.5(b). Also shown in Figure 10.5(b), convergence of the lower-level fitness function does not exhibit a monotonically-increasing trend, but rather demonstrates certain coincidence with the trend of the upper-level fitness function as shown in Figure 10.5(a). This suggests that the upper-level decisions do affect convergence of the lower-level fitness function, consistent with the coupling of two different optimization problems at two levels. Therefore, the lower-level optimal solution must conform to the achievement of the optimization goal at the upper level. In other words, satisfying the upper-level optimization is the premise of finding a solution to the lower-level problem.



(a) Upper-level iteration process



(b) Lower-level iteration process

Figure 10.5 Convergence of optimal solutions by GA fitness performance

10.4 System Layout Design

The system layout design is based on a weighted process flow. The weight reflects the clustering tendency of the nearby two processes. Generally speaking, the process with slower setup time and process time tends to become the bottle neck in an assembly system, so it should be located in a cluster closer to nearby work stations rather than placed to some remote location. So the weight rating between each process can be calculated as following:

$$W(i,j) = \frac{(T_i^{SET} + T_j^{SET})(T_i^{p} + T_j^{p})}{Max((T_n^{SET} + T_{n+1}^{SET})(T_n^{p} + T_{n+1}^{p}))}, n \in (1,23) . \qquad (10.2)$$

Based on the calculation, I can have a weighted process flow reflecting the overall system material flow and their relative importance. Then it will be converted to a DSM for cluster calculation. The weighted process flow and corresponding DSM are as shown in Figure 10.6.

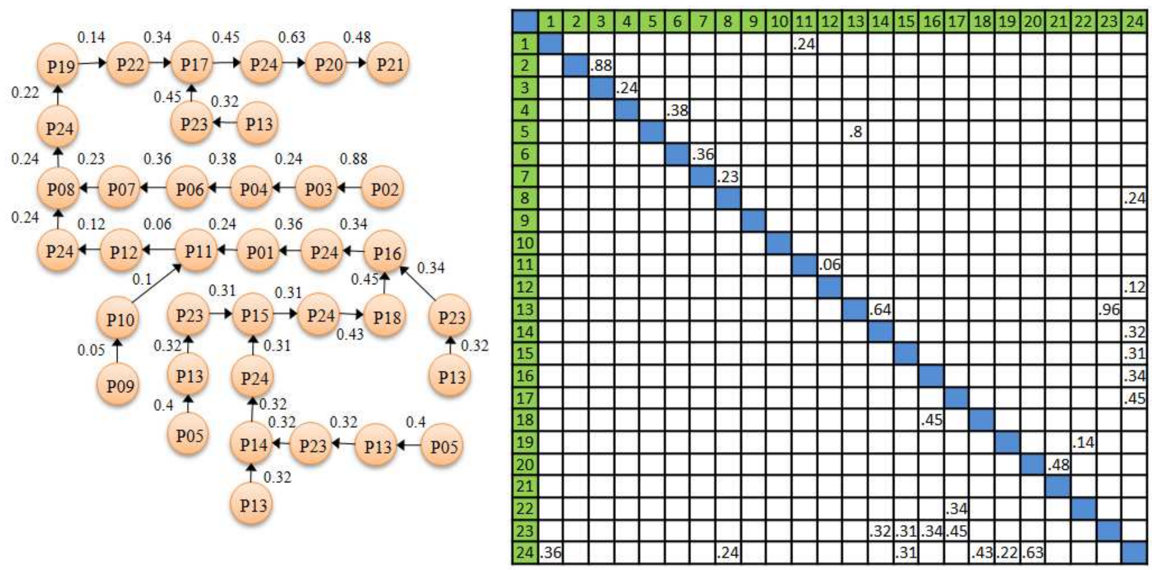


Figure 10.6 Weighted assembly process flow and its DSM

After clustering analysis, the result is shown in Figure 10.7. There are totally 10 clusters identified with coordination cost of 122. In the clustering result: Process 8, 15, 20, 21, and 24 are in cluster 1; Process 5, 13, 14, 17, and 23 are in cluster 2; Process 4, 6, and 7 are in cluster 3; Process 19 and 22, 16 and 18, 2 and 3, 1 and 11 are in cluster 4 to 7; The rest processes including 12, 10, and 9 are inside a cluster by themselves.

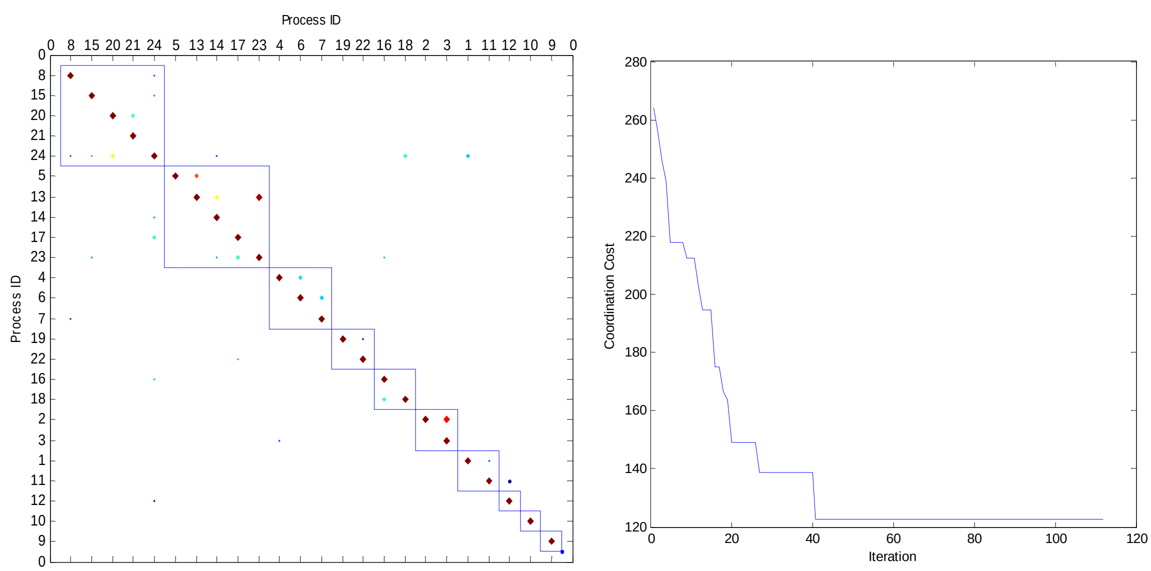


Figure 10.7 Weighted assembly process flow and its DSM

It is obvious, the cluster 1 includes the most important assembly processes and final output, including connector housing assemble and soldering. cluster 2 handle the additional assembly parts such as sealing and latch. cluster 3 is a wire handling cluster, and the rest clusters are very dedicated clusters such as wire pre-process, pin handling, stitching and bending, solder inspection and test clusters. The clustering analysis shows a very reasonable result.

10.5 Data Driven Simulation

The assembly system factory floor setup can be determined after the process design and layout design are completed. As shown in Figure 10.8, the clusters identified using DSM can be seen in the factory floor setup. Some machines are integrated into one flexible work station to gain factory space utilization.

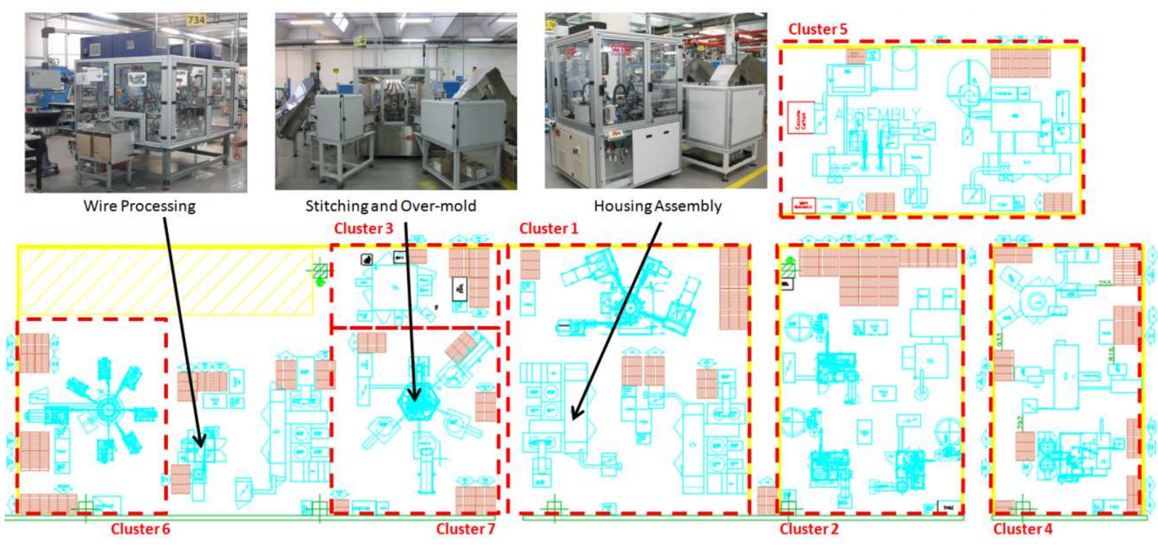


Figure 10.8 Connector assembly system physical plant setup

System running data is gathered using industrial PC in each workstation which is connected with sensors, PLC, machine vision cameras, RFID readers, and robot controllers. As shown in Figure 10.9, each workstation has its own industrial PC as the brain to collect data as well as sending control commands. It will have either one way (data input only) or two way communication (data input and control) with all downstream equipments. Then data package from industrial PC of each workstation will be sent to server which has the online simulation tools setup.

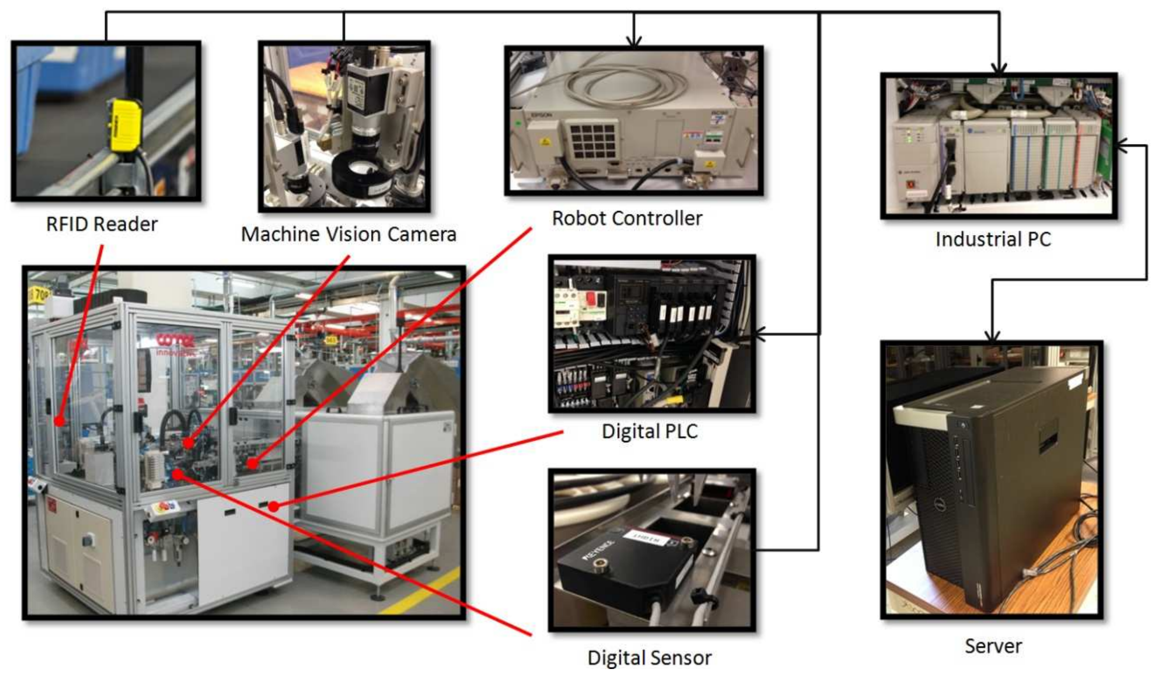


Figure 10.9 Data collection and control feedback infrastructure

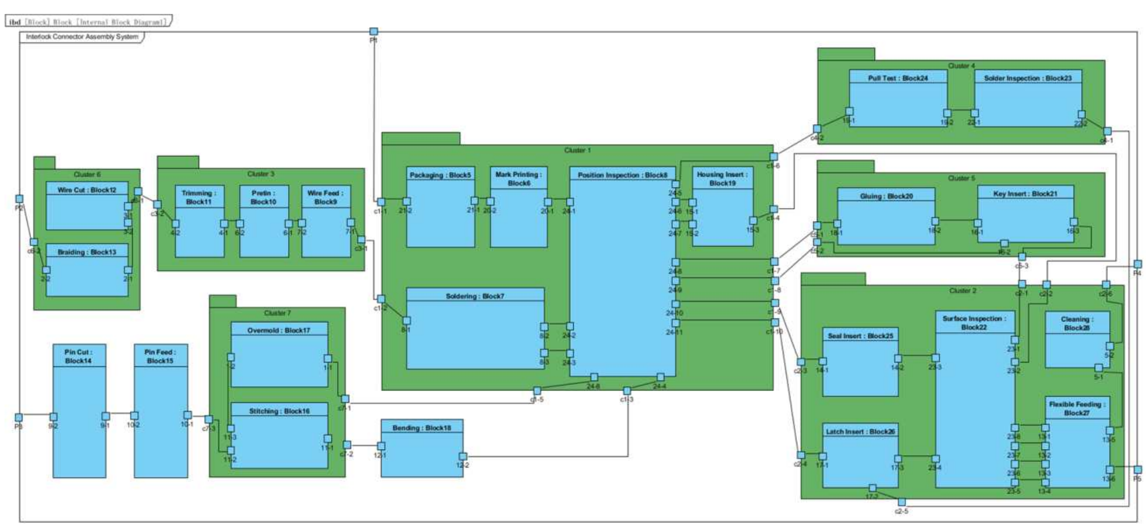


Figure 10.10 SysML model of the connector assembly system

In the online simulation tool, a generic parameterized SysML model is used to construct the Arena model. For each given process design set *x* and resource allocation plan *y*, there is a corresponding simulation model which is automatically converted from the parameterized SysML model. The total number of possible simulation models will be over a hundred in this case study, due to the huge demand to variety of this connector product family. In this chapter, just one SysML model driven simulation of the process designs will be discussed. As the current physical platform of this connector assembly system is still under construction, the real-time data collection is limited within several workstations.

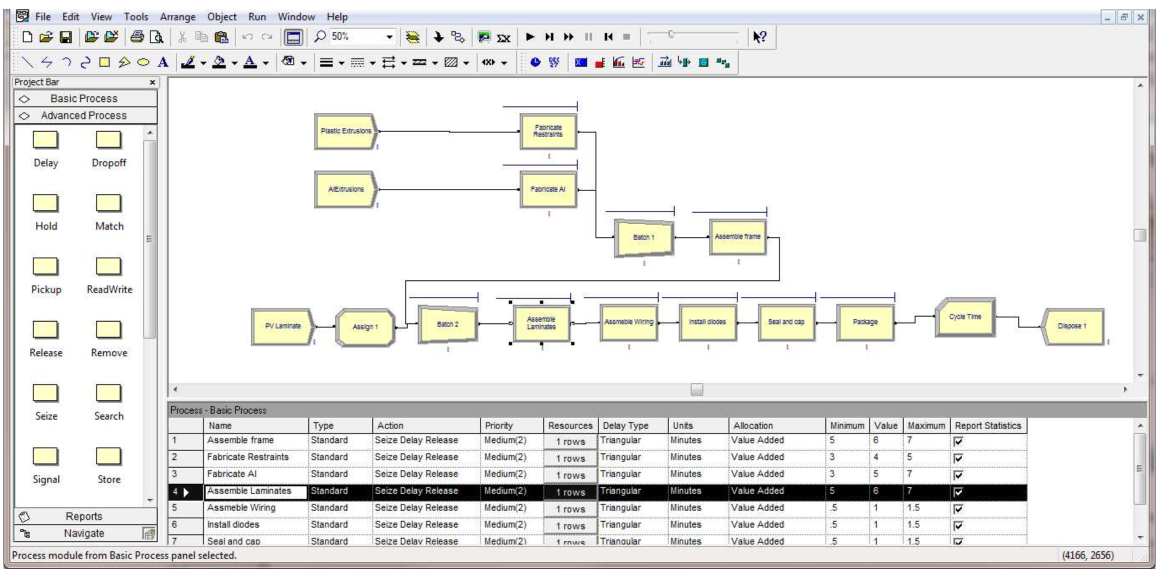
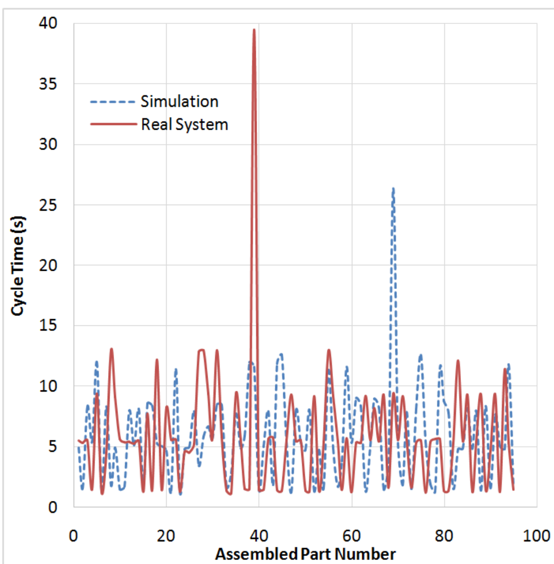


Figure 10.11 Arena simulation model for one of the assembly plan set

A comparison between the simulation result and actual workstation test data is shown in Figure 10.12. The test data is acquired from a field test of cluster 1 and 2, including flexible feeding, key insertion and housing insertion. The cycle time is recorded in cluster 1's PC and uploaded to the server. The simulation used same assembly plan set setup and machine parameters. Different from the traditional delicate die tooling assembly system, which normally has a consistent cycle time, the high variety assembly system equipped with flexible units such as flexible feeders and robots, working on mixed products would have a stochastic cycle time. The left chart shows the individual cycle time in the 100 parts run plotted in original order, the cycle time varies from 2 to 40 seconds. The chart on the right side shows the same run with all individual cycle time ranked from low to high. Both the real system recording and simulation shows the character time steps caused by mixed part in flexible material handling in assembly process. The average cycle time of real system is 5.61s where as simulation result averages at 5.72s. The variance of cycle time difference is less than 2%.



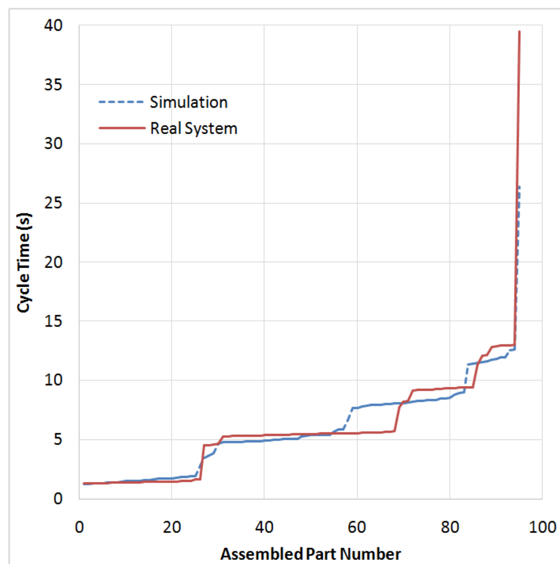


Figure 10.12 Comparison of simulation result and real system

This product family was handled by mixed assembly line of human manual assembly and delicate machines before the design and construction of the new automatic high variety assembly system. The average cycle time for housing insertion of the pervious system is 7.5s, the average system OEE is 76%. As a comparison, the new system using the optimized assembly process plan and system setup, has an average cycle time of 5.61s and 81% OEE, which is 25.2% and 6.6% improvement.

10.6 Discussion

The case study of a connector's high variety assembly system design shows an example of how much is the influence that a product variety would bring to the assembly system design. This case of seven-part assembly is just one product family from the whole span of automotive interlock connectors. With totally 864 varieties coming from combination of 79 parts, it already exceeds the limit of traditional assembly system when producing the whole product family form one system is desired.

The game theoretic optimization shows a powerful capability of delivering a reasonably high quality solution to this high variety assembly system design problem. Together with system layout clustering analysis and simulation verification, it shows the potential to help field engineers to solve the becoming more and more complicated decision problems in high variety assembly system design.

10.7 Summary

The original achievements in this chapter can be concluded as followings:

1) The whole process of using the proposed game theoretic optimization method to design and optimize a high variety connector assembly system is demonstrated, it can be reused on other similar high variety production systems;
2) The game theoretic optimization of assembly system design and layout design is validated through the case study;

3) The solution is provided to the design and optimization problem in applications of newly developed flexible assembly system, which is a current challenge met by many companies when shifting from traditional manufacturing system to flexible manufacturing system.

The case study of a connector's high variety assembly system design served as a validation of the methodology set proposed in this dissertation. By going into the detailed design decision making based on the given connector family, it shows the actual working process of the proposed high variety assembly system design framework and its game theoretic optimization solution, as well as providing a validation to the proposed game theoretic optimization solution to assembly system design.

CHAPTER 11

CONCLUSIONS AND FUTURE WORK

This concluding chapter summarizes the findings and the contributions of the thesis work. The limitations and possible improvements are also discussed, along with avenues for future research.

11.1 Conclusions

The mass customization paradigm has brought fundamental change to the way a product is designed and manufactured. Continuous growing demands of customized products, increasing competition among manufacturing industries and increasing labor cost are all demanding mass customization to be realized in all spectrum of industry. With the improvement of technologies, such as robotics, machine vision, flexible material handling, and simulation tools, the chance to bring mass customization into diverse industrial applications has come. The assembly system, being identified as the breaking point to enable mass customization, also bring challenges when dealing with high variety products which is typical situation in mass customization. In order to tackle such a complex design problem systematically, a technical framework is proposed, dealing with variety formulation, high variety assembly system layout, leveraging between assembly process design and resource allocation, and assembly process planning.

First, in order to identify the necessary process elements and their relations for a given product variety demands, the mapping between the product variety to process variety must be formulated. By using a generic representation of both product and process information, the large amount of variety data of both product and process can be handled. Then the construction of association rules mining makes it possible to find a suitable assembly process set to deliver the process variety that can fulfill the product variety demand. Then the transformation from a product definition to a feasible process set achieving the required variety can be preformed.

Second, the major decision making problem underlying the high variety assembly system design problem is the equilibrium solution between the assembly process design and resource allocation. With the evaluation criteria of assembly flexibility and utilization rate, the assembly system design decision making problem becomes a leverage between flexibility and efficiency of the system design. The game theoretic optimization framework together with LFGA brings a mathematical solution to this problem.

Third, the physical design of assembly system is the foundation of the of high variety assembly system realization. It is an important constraint to the assembly system resource allocation and process planning. The task of designing an assembly system layout can be generally concluded as grouping similar and highly dependent assembly processes together into an enclosed unit, a department or a work station, depending on their scale. The use of DSM enables the whole layout design process to be automatic, by using a coordination cost based clustering algorithm, a given assembly process set can be translated into a system layout which will optimize the material flow efficiency and minimize possible bottlenecks in the system.

Fourth, in order to verify and validate the assembly system design, a real-time model driven simulation method and an industrial application case study is reported. The simulation brings not only a methodology to verify the system design, but also a possible industrial application on online simulation and feedback to keep improvement to the assembly system. The case study as a validation to the methodology set proposed in assembly system design, illustrates the process flow to solve a real life high variety assembly system design problem, and also proves the necessity of proposing a feasible methodology to solve the problem in assembly system design brought by high variety. Through the field application of game theoretic optimization, its advantage on providing improvements to complex assembly system design is validated.

11.2 Contributions

The major contributions of the dissertation manifest themselves through the proposal and development of a game theoretic optimization framework for high variety assembly system design. The contributions are elaborated below:

(1) **Game theoretic optimization method for complex engineering system design decision making**: The complex engineering system with competing design decision making problems are formulated as a game theoretic optimization problem, which uses bi-level decision making model based on non-cooperative game instead of rationally used all-in-one solution. The widely used genetic algorithm is also modified to provide a numerical solution to the game theoretic optimization method;

(2) S**ystematic solution framework for high variety assembly system design to achieve mass customization**: The high variety assembly system design problem is formulated as a game theoretic optimization problem. The framework to solve this problem includes new evaluation criteria (process commonality evaluation and resource utilization evaluation) which serves as fitness function for both leader and follower in the

non-cooperative game, modified GA solution and association rule mining method to identify the relationship of generic product variety and process variety process. It is validated by the case study and shows a reasonable efficiency to solve the high variety assembly system design problem;

(3) **First practice in industry on high variety assembly system design based on mixed traditional and newly developed flexible manufacturing systems**: Different from fixed die assembly systems, the design and development of high variety assembly system based on flexible manufacturing system is carried out with coordination cost function and setup time / process time weighted assembly process based DSM clustering, which is then used as reference of the physical assembly system layout design. Online data acquisition and real-time simulation based decision making feedback architecture are used to handle the highly dynamic high variety assembly process planning problem after the physical system is built. The design and optimization problem in applications of newly developed flexible assembly system, which is a current challenge met by many companies when shifting from traditional manufacturing system to flexible manufacturing system, is practiced in this research and also used as a validation to the proposed game theoretic optimization framework.

11.3 Limitations

As an exploratory study of the proposed game theoretic optimization of high variety assembly system design, it suffers several limitations.

(1) Evaluation criteria and equilibrium solution: In this dissertation, I use a GA based global stochastic search algorithm to identify improvements over each iteration of game theoretic optimization. Different evaluation criteria would lead to a different equilibrium solution on the same given feasible region. The evaluation criteria I used are based on a comprehensive study from multiple research result in this field over the last several decades. However, as the evaluation method is always using simplification and abstraction of the real system, only the key concern in the assembly system design is covered. A real life application of the assembly system design will always have more concerns and limitations which will affect the decision making of design and planning. In this solution framework, the major concern is focused on the system flexibility and equipment utilization to allow all products in same family to be assembled on given set of equipments. In an industrial application, there are much more concerns such as leverage between performance and flexibility, return of investment of the system, product lifecycle and so on. So the solution has a limited level of fidelity to reflect the requirement from a real system. Although it is beyond the scope to provide a full evaluation criteria set that covers all possible assembly system industrial applications, the given example of process commonality and resource utilization evaluation demonstrates how these evaluation criteria are formulated. So it would be meaningful to be used as an application example and modified based on any further application practices.

(2) Game theoretic optimization application: The game theoretic optimization is proposed to provide solution for different kinds of complex engineered systems' decision making. However, in this dissertation, the scope of its application is limited to find the equilibrium solution to the assembly process design and resource allocation problem. Actually there are more than one competing goals in the design task of a complex system, such is high variety assembly system design. The overall cycle time, OEE, cost, return of investment, system reuse in end of life cycle, and reliability are all competing goals that could use the game theoretic optimization to find an equilibrium solution. It is beyond the scope to find all possible competing goals in an assembly system design problem. This dissertation picks the process design and resource allocation problem as a representative problem in assembly system design to present the capability of game theoretic optimization.

(3) Simulation tool application: The proposed online simulation framework is limited in application by the constraint of hardware availability. Factories have been using information technology to collect production data for years, but it takes a lot of efforts to summarize all kinds of data based on different hardware systems and protocols. The real life application of online simulation would require a team effort of engineers from different expertise to be realized. Based on the simulation tools, it is also possible to provide a simulation based optimization to assist the assembly system design. However, the computational load on such application limited such practice in this dissertation.

11.4 Future Work

The realization of mass customization will totally change the way of design and manufacturing of the whole industry. Several ideas are elaborated below for potential endeavors in the future.

(1) Game theory based optimization with multiple parties: The original game theory model used in this dissertation involves two parties called the leader and follower, which comes from a simplified business competition. The real life engineering problems always have more than two parties with different goals that looking for an equilibrium solution. The topic of high variety assembly system design already suggested more than two competing goals exists. The exploration of an optimization framework leveraging multiple parties with different interests based on game theory will be a topic worth to study. It will bring a more realistic representation of the actual problem in engineering practice.

(2) Simulation integrated application in factory floor: The development of big data and digital factory has brought huge quantity of data available to be used. However, there is still no effective way to utilize these data on helping continuous improvement to the system in a timely manner. Traditional ways of simulation technology are still limited to offline mode which is becoming too slow for some applications. The real-time simulation tool proposed in this research shows great potential but still need further development on integration to the manufacturing system to the industrial standard. With a fully developed online simulation integrated system, it will bring a new way on operation excellence in manufacturing.

REFERENCES

Adham, A. A., & Tahar, R. B. M. (2012). Optimum efficiency of production line in automobile manufacture. African Journal of Business Management, 6(20), 6266-6275.

Agard, B., & Kusiak*, A. (2004). Data-mining-based methodology for the design of product families. International Journal of Production Research,42(15), 2955-2969.

Alyaqout, S. F., Peters, D. L., Papalambros, P. Y., & Ulsoy, A. G. (2011). Generalized coupling management in complex engineering systems optimization. Journal of Mechanical Design, 133(9), 091005.

Anandalingam, G., & White, D. J. (1990). A solution method for the linear static Stackelberg problem using penalty functions. Automatic Control, IEEE Transactions on, 35(10), 1170-1173.

Baker, J. E. (1987, July). Reducing bias and inefficiency in the selection algorithm. In Proceedings of the second international conference on genetic algorithms (pp. 14-21).

Baldwin, R. A., & Chung, M. J. (1995). Managing engineering data for complex products. Research in Engineering Design, 7(4), 215-231.

Banerjee, P., & Zhou, Y. (1995). Facilities layout design optimization with single loop material flow path configuration. THE INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH, 33(1), 183-203.

Batarseh, O., & McGinnis, L. F. (2012, December). System modeling in sysml and system analysis in arena. In Proceedings of the Winter Simulation Conference (p. 258). Winter Simulation Conference.

Bateman, R. J., & Cheng, K. (2002). Devolved manufacturing. Concurrent Engineering, 10(4), 291-298.

Bi, Z. M., Gruver, W. A., & Lang, S. Y. (2004). Analysis and synthesis of reconfigurable robotic systems. Concurrent Engineering, 12(2), 145-153.

Bialas, W. F., & Karwan, M. H. (1982). On two-level optimization. Automatic Control, IEEE Transactions on, 27(1), 211-214.

Bialas, W. F., & Karwan, M. H. (1984). Two-level linear programming.Management science, 30(8), 1004-1020.

Bikas, C., Argyrou, A., Pintzos, G., Giannoulis, C., Sipsas, K., Papakostas, N., & Chryssolouris, G. (2016). An Automated Assembly Process Planning System. Procedia CIRP, 44, 222-227.

Bourjault, A. (1984). Contribution to a methodological approach of automated assembly: automatic generation of assembly sequence. University de Franche-Comte.

Bracken, J., & McGill, J. T. (1973). Mathematical programs with optimization problems in the constraints. Operations Research, 21(1), 37-44.

Bukchin, Y., Meller, R. D., & Liu, Q. (2006). Assembly system facility design. IIE Transactions, 38(1), 53-65.

Canen, A. G., & Williamson, G. H. (2013). Facility layout overview: towards competitive advantage. Facilities.

Chen, Z., & Wang, L. (2010). Personalized product configuration rules with dual formulations: A method to proactively leverage mass confusion. Expert Systems with Applications, 37(1), 383-392.

Chhajed, D., Montreuil, B., & Lowe, T. J. (1992). Flow network design for manufacturing systems layout. European Journal of Operational Research,57(2), 145-161.

Chien, T. K. (2004). An empirical study of facility layout using a modified SLP procedure. Journal of Manufacturing Technology Management, 15(6), 455-465.

Chittratanawat, S. (1999). An integrated approach for facility layout, P/D location and material handling system design. International Journal of Production Research, 37(3), 683-706.

Choi, C. K., Zha, X. F., Ng, T. L., & Lau, W. S. (1998). On the automatic generation of product assembly sequences. International Journal of Production Research, 36(3), 617-633.

Culberson, J. C. (1998). On the futility of blind search: An algorithmic view of “no free lunch”. Evolutionary Computation, 6(2), 109-127.

Dai, Z., & Scott, M. J. (2007). Product platform design through sensitivity analysis and cluster analysis. Journal of Intelligent Manufacturing, 18(1), 97-113.

Dai, Z., & Scott, M. J. (2007). Product platform design through sensitivity analysis and cluster analysis. Journal of Intelligent Manufacturing, 18(1), 97-113.

Darwin, C. (1963). The Origin of Species Or, The Preservation of Favoured Races in the Struggle for Life. J. W. Burrow (Ed.). Emereo.

De Fazio, T. L., & Whitney, D. E. (1987). Simplified generation of all mechanical assembly sequences. Robotics and Automation, IEEE Journal of,3(6), 640-658.

De Lit, P., Danloy, J., Delchambre, A., & Henrioud, J. M. (2003). An assembly-oriented product family representation for integrated design. Robotics and Automation, IEEE Transactions on, 19(1), 75-88.

De Mello, H., & Scaramelli, L. (1989). Task sequence planning for robotic assembly.

De Mello, L. S. H., & Sanderson, A. C. (1991). A correct and complete algorithm for the generation of mechanical assembly sequences. Robotics and Automation, IEEE Transactions on, 7(2), 228-240.

De Mello, L. S. H., & Sanderson, A. C. (1991). Representations of mechanical assembly sequences. Robotics and Automation, IEEE Transactions on, 7(2), 211-227.

DeJong, K. (1975). An analysis of the behavior of a class of genetic adaptive systems. Ph. D. Thesis, University of Michigan.

Devendorf, E., & Lewis, K. (2011). The impact of process architecture on equilibrium stability in distributed design. Journal of Mechanical Design,133(10), 101001.

Dewa, M., & Chidzuu, L. (2012, June). Bottleneck Management Through Discrete Event Simulation for Manual Automobile Assembly Systems. InComputers and Industrial Engineering 42.

Dini, G., Failli, F., Lazzerini, B., & Marcelloni, F. (1999). Generation of optimized assembly sequences using genetic algorithms. CIRP Annals-Manufacturing Technology, 48(1), 17-20.

Du, J., Jiao, Y. Y., & Jiao, J. (2005). Integrated BOM and routing generator for variety synchronization in assembly-to-order production. Journal of Manufacturing Technology Management, 16(2), 233-243.

Du, X., Jiao, J., & Tseng, M. (2003). Modelling platform-based product configuration using programmed attributed graph grammars. Journal of Engineering Design, 14(2), 145-167.

Du, X., Jiao, J., & Tseng, M. M. (2001). Architecture of product family: fundamentals and methodology. Concurrent Engineering, 9(4), 309-325.

Eastman, C. M. (1981). The design of assemblies (No. 810197). SAE Technical Paper.

Edmunds, T. A., & Bard, J. F. (1992). An algorithm for the mixed-integer nonlinear bilevel programming problem. Annals of Operations Research, 34(1), 149-162.

Eppinger, S. D., & Ulrich, K. T. (1995). Product design and development.Product design and development.

Fan, Y. S., & Huang, G. Q. (2007). Networked manufacturing and mass customization in the ECommerce era: the Chinese perspective.

Fellini, R., Kokkolaras, M., Papalambros, P. Y., & Perez-Duarte, A. (2002, January). Platform selection under performance loss constraints in optimal design of product families. In ASME 2002 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (pp. 613-621). American Society of Mechanical Engineers.

Ferguson, S. M., Olewnik, A. T., & Cormier, P. (2014). A review of mass customization across marketing, engineering and distribution domains toward development of a process framework. Research in Engineering Design, 25(1), 11-30.

Fieldbook, W. (2004). The Toyota Way Fieldbook.

Fisher, M., Ramdas, K., & Ulrich, K. (1999). Component sharing in the management of product variety: A study of automotive braking systems.Management Science, 45(3), 297-315.

Fonseca, C. M., & Fleming, P. J. (1995). An overview of evolutionary algorithms in multiobjective optimization. Evolutionary computation, 3(1), 1-16.

Ford, H. (2007). My life and work. Cosimo, Inc..

Fortuny-Amat, J., & McCarl, B. (1981). A representation and economic interpretation of a two-level programming problem. Journal of the operational Research Society, 783-792.

Francis, R. L., McGinnis, L. F., & White, J. A. (1992). Facility layout and location: an analytical approach. Pearson College Division.

Franz, C., Koberstein, A., & Suhl, L. (2015). Dynamic resequencing at mixed-model assembly lines. International Journal of Production Research,53(11), 3433-3447.

Friedenthal, S., Moore, A., & Steiner, R. (2014). A practical guide to SysML: the systems modeling language. Morgan Kaufmann.

Garey, M. R., & Johnson, D. S. (1979). Computers and intractability: a guide to the theory of NP-completeness. 1979. San Francisco, LA: Freeman.

Gen, M., & Cheng, R. (2000). Genetic algorithms and engineering optimization (Vol. 7). John Wiley & Sons.

Golberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. Addion wesley, 1989, 102.

Gupta, S., & Krishnan, V. (1998). Product family-based assembly sequence design methodology. IIE transactions, 30(10), 933-945.

Haddadzade, M., Razfar, M. R., & Farahnakian, M. (2009). Integrating process planning and scheduling for prismatic parts regard to due date.World Academy of Science, Engineering and Technology, 51, 64-67.

Hansen, P., Jaumard, B., & Savard, G. (1992). New branch-and-bound rules for linear bilevel programming. SIAM Journal on scientific and Statistical Computing, 13(5), 1194-1217.

Harding, J. A., Shahbaz, M., & Kusiak, A. (2006). Data mining in manufacturing: a review. Journal of Manufacturing Science and Engineering,128(4), 969-976.

Hazelrigg, G. A. (1998). A framework for decision-based engineering design.Journal of mechanical design, 120(4), 653-658.

Heemskerk, I. C., & Van Luttervelt, C. A. (1989). The use of heuristics in assembly sequence planning. CIRP Annals-Manufacturing Technology, 38(1), 37-40.

Hegge, H. M. H., & Wortmann, J. C. (1991). Generic bill-of-material: a new product model. International Journal of Production Economics, 23(1), 117-128.

Hellenbrand, D., & Lindemann, U. (2008). Using the DSM to support the selection of product concepts. In DSM 2008: Proceedings of the 10th International DSM Conference, Stockholm, Sweden, 11.-12.11. 2008.

Heragu, S. S. (2008). Facilities design. CRC Press.

Hölttä-Otto, K., Tang, V., & Otto, K. (2008). Analyzing module commonality for platform design using dendrograms. Research in Engineering Design, 19(2-3), 127-141.

Hopp, W. J., & Spearman, M. L. (2011). Factory physics. Waveland Press.

Hu, S. J., Ko, J., Weyand, L., ElMaraghy, H. A., Lien, T. K., Koren, Y., ... & Shpitalni, M. (2011). Assembly system design and operations for product variety. CIRP Annals-Manufacturing Technology, 60(2), 715-733.

Hu, S., & Linwei, X. (2009, May). Optimization of scheduling problem for auto mixed model assembly line. In Intelligent Systems and Applications, 2009. ISA 2009. International Workshop on (pp. 1-3). IEEE.

Huang, E., Ramamurthy, R., & McGinnis, L. F. (2007, December). System and simulation modeling using SysML. In Proceedings of the 39th conference on Winter simulation: 40 years! The best is yet to come (pp. 796-803). IEEE Press.

Huang, Y. F., & Lee, C. G. (1989, May). Precedence knowledge in feature mating operation assembly planning. In Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on (pp. 216-221). IEEE.

Huang, Y., Liu, H., Keong Ng, W., Lu, W., Song, B., & Li, X. (2008). Automating knowledge acquisition for constraint-based product configuration.Journal of Manufacturing Technology Management, 19(6), 744-754.

Irudayaraj, P., Liu, Y., Zhou, F., Jiao, R. J., & Goodman, J. N. (2014, August). A Case Study of Modeling and Simulation for Manufacturing, Installation, and Maintenance of Solar Power Systems. In ASME 2014 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (pp. V004T06A033-V004T06A033). American Society of Mechanical Engineers.

Jain, A., Jain, P. K., & Singh, I. P. (2006). An integrated scheme for process planning and scheduling in FMS. The International Journal of Advanced Manufacturing Technology, 30(11-12), 1111-1118.

Jiao, J. R., Simpson, T. W., & Siddique, Z. (2007). Product family design and platform-based product development: a state-of-the-art review. Journal of intelligent Manufacturing, 18(1), 5-29.
Jiao, J., & Tseng, M. M. (1999). A methodology of developing product family architecture for mass customization. Journal of Intelligent Manufacturing, 10(1), 3-20.
Jiao, J., & Tseng, M. M. (1999). An information modeling framework for product families to support mass customization manufacturing. CIRP Annals-Manufacturing Technology, 48(1), 93-98.
Jiao, J., & Tseng, M. M. (2000). Fundamentals of product family architecture. Integrated Manufacturing Systems, 11(7), 469-483.
Jiao, J., & Tseng, M. M. (2000). Understanding product family for mass customization by developing commonality indices. Journal of Engineering Design, 11(3), 225-243.
Jiao, J., Tseng, M. M., Duffy, V. G., & Lin, F. (1998). Product family modeling for mass customization. Computers & Industrial Engineering, 35(3), 495-498.
Jiao, J., Tseng, M. M., Ma, Q., & Zou, Y. (2000). Generic bill-of-materials-and-operations for high-variety production management. Concurrent Engineering,8(4), 297-321.
Jiao, J., Zhang, L., & Pokharel, S. (2007). Process platform planning for variety coordination from design to production in mass customization manufacturing. Engineering Management, IEEE Transactions on, 54(1), 112-129.
John, H. (1992). Holland, Adaptation in natural and artificial systems.
Kanai, S., Takahashi, H., & Makino, H. (1996). ASPEN: computer-aided assembly sequence planning and evaluation system based on predetermined time standard. CIRP Annals-Manufacturing Technology, 45(1), 35-39.
Kang, S., Gong, J., & Ume, I. C. (2011, January). Determination of optimum values of control parameters to reduce laser speckle noise for projection moiré system. In ASME 2011 International Mechanical Engineering Congress and Exposition (pp. 743-749). American Society of Mechanical Engineers.
Kelkar, A., & Koc, B. (2008). Geometric planning and analysis for hybrid re-configurable molding and machining process. Rapid Prototyping Journal, 14(1), 23-34.
Khosla, P. K., & Mattikali, R. (1989). Determining the assembly sequence from a 3-D model. Journal of Mechanical Working Technology, 20, 153-162.
Kim, J. G., & Kim, Y. D. (2000). Layout planning for facilities with fixed shapes and input and output points. International Journal of Production Research,38(18), 4635-4653.
Kim, K. H., & Egbelu, P. J. (1999). Scheduling in a production environment with multiple process plans per job. International Journal of Production Research, 37(12), 2725-2753.
Kim, S. C., & Bobrowski, P. M. (1994). Impact of sequence-dependent setup time on job shop scheduling performance. The International Journal of Production Research, 32(7), 1503-1520.
Kokkolaras, M., Mourelatos, Z. P., & Papalambros, P. Y. (2004, January). Design optimization of hierarchically decomposed multilevel systems under uncertainty. In ASME 2004 International Design Engineering Technical Conferences and

Computers and Information in Engineering Conference (pp. 613-624). American Society of Mechanical Engineers.

Kota, S., Sethuraman, K., & Miller, R. (2000). A metric for evaluating design commonality in product families. Journal of Mechanical Design, 122(4), 403-410.

Lee, K., & Gossard, D. C. (1985). A hierarchical data structure for representing assemblies: part 1. Computer-Aided Design, 17(1), 15-19.

Leung, C. W., Wong, T. N., Mak, K. L., & Fung, R. Y. (2010). Integrated process planning and scheduling by an agent-based ant colony optimization.Computers & Industrial Engineering, 59(1), 166-180.

Li, S., Wang, H., Hu, S. J., Lin, Y. T., & Abell, J. A. (2011). Automatic generation of assembly system configuration with equipment selection for automotive battery manufacturing. Journal of Manufacturing Systems, 30(4), 188-195.

Liu, Y. T., Wang, C. J., & Wang, B. (2011). Design and Development on the Processing and Managing System of Spaceflight Products Analysis and Experimental Data (Vol. 42, pp. 299-302). Trans Tech Publications.

Liu, Y., Ji, Y., & Jiao, R. J. (2013). A Stackelberg Solution to Joint Optimization Problems: A Case Study of Green Design. Procedia Computer Science, 16, 333-342.

Luggen, W. W. (1991). Flexible manufacturing cells and systems. Prentice Hall.

Luo, L. (2011). Product line design for consumer durables: an integrated marketing and engineering approach. Journal of Marketing Research, 48(1), 128-139.

Martín-Fernández, J. A., Barceló-Vidal, C., & Pawlowsky-Glahn, V. (1998). Measures of difference for compositional data and hierarchical clustering methods. In A. Buccianti, G. Nardi, & R. Potenza (Eds.), Proceedings of IAMG (Vol. 98, pp. 526-531).

Mather, H. (1987). Bills of materials. Irwin Professional Pub.

Mathieu, R., Pittard, L., & Anandalingam, G. (1994). Genetic algorithm based approach to bi-level linear programming. Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle, 28(1), 1-21.

McGinnis, L., & Ustun, V. (2009, December). A simple example of SysML-driven simulation. In Proceedings of the 2009 Winter Simulation Conference (WSC) (pp. 1703-1710). IEEE.

Meyer, M. H., & Lehnerd, A. P. (1997). The power of product platforms. Simon and Schuster.

Michalos, G., Makris, S., Papakostas, N., Mourtzis, D., & Chryssolouris, G. (2010). Automotive assembly technologies review: challenges and outlook for a flexible and adaptive approach. CIRP Journal of Manufacturing Science and Technology, 2(2), 81-91.

Mills, S. F., & Tanik, M. M. (2000). Resource-focused process engineering for the distributed enterprise. International Journal of Computer Integrated Manufacturing, 13(3), 187-203.

Montreuil, B. (1991). A modelling framework for integrating layout design and flow network design. In Material Handling’90 (pp. 95-115). Springer Berlin Heidelberg.

Montreuil, B., & Venkatadri, U. (1991). Strategic interpolative design of dynamic manufacturing systems layouts. Management Science, 37(6), 682-694.

Mourtzis, D., Doukas, M., & Psarommatis, F. (2014). A toolbox for the design, planning and operation of manufacturing networks in a mass customisation environment. Journal of Manufacturing Systems.

Mourtzis, D., Doukas, M., Vlachou, A., & Xanthopoulos, N. (2014). Machine Availability Monitoring for Adaptive Holistic Scheduling: A Conceptual Framework for Mass Customization. Procedia CIRP, 25, 406-413.

Müller, R., Esser, M., & Vette, M. (2013). Reconfigurable handling systems as an enabler for large components in mass customized production. Journal of Intelligent Manufacturing, 24(5), 977-990.

Muther, R. (1973). Systematic Layout Planning/by Richard Muther (No. 658.5 m8.).

Nicholls, M. G. (1996). The application of non-linear bi-level programming to the aluminium industry. Journal of Global Optimization, 8(3), 245-261.

Nishizaki, I., & Sakawa, M. (1999). Stackelberg solutions to multiobjective two-level linear programming problems. Journal of Optimization Theory and Applications, 103(1), 161-182.

Object Management Group (OMG). Model driven architecture. Version 1.0.1. www.omg.org/cgi-bin/doc?omg/03-06-01.pdf (2003).

Object Management Group (OMG). SysML–Modelica transformation (SyM). www.omg.org/spec/SyM/1.0/PDF/ (2012).

Ojaghi, Y., Khademi, A., Yusof, N. M., Renani, N. G., & bin Syed Hassan, S. A. H. (2015). Production Layout Optimization for Small and Medium Scale Food Industry. Procedia CIRP, 26, 247-251.

Olsen, K. A., Sætre, P., & Thorstenson, A. (1997). A procedure-oriented generic bill of materials. Computers & industrial engineering, 32(1), 29-45.

Osborne, M. J., & Rubinstein, A. (1994). A course in game theory. MIT press.

Pine, B. J. (2000). Markets of one: creating customer-unique value through mass customization. Harvard Business Press.

Pirlot, M. (1992). General local search heuristics in combinatorial optimization: a tutorial. Belgian Journal of Operations Research, Statistics and Computer Science, 32(1-2), 7-69.

Randall, T., & Ulrich, K. (2001). Product variety, supply chain structure, and firm performance: Analysis of the US bicycle industry. Management Science, 47(12), 1588-1604.

Rane, A. B., Sudhakar, D. S. S., Sunnapwar, V. K., & Rane, S. (2015, January). Improving the performance of assembly line: Review with case study. In Nascent Technologies in the Engineering Field (ICNTE), 2015 International Conference on (pp. 1-14). IEEE.

Rekiek, B., & Delchambre, A. (2006). Assembly line design: the balancing of mixed-model hybrid assembly lines with genetic algorithms. Springer Science & Business Media.

Robertson, D., & Ulrich, K. (1998). Planning for product platforms. MIT Sloan Management Review, 39(4), 19.

Roy, R., Hinduja, S., & Teti, R. (2008). Recent advances in engineering design optimisation: Challenges and future trends. CIRP Annals-Manufacturing Technology, 57(2), 697-715.

Roy, U., & Liu, C. R. (1988). Establishment of functional relationships between product components in assembly database. Computer-Aided Design, 20(10), 570-580.

Saygin, C., & Kilic, S. E. (1999). Integrating flexible process plans with scheduling in flexible manufacturing systems. The International Journal of Advanced Manufacturing Technology, 15(4), 268-280.

Šekoranja, B., Bašić, D., Švaco, M., Šuligoj, F., & Jerbić, B. (2014). Human-robot interaction based on use of capacitive sensors. Procedia Engineering,69, 464-468.

Shabde, V. S., & Hoo, K. A. (2008). Optimum controller design for a spray drying process. Control Engineering Practice, 16(5), 541-552.

Shah, R., & Ward, P. T. (2003). Lean manufacturing: context, practice bundles, and performance. Journal of operations management, 21(2), 129-149.

Shahbaz, M., Srinivas, M., Harding, J. A., & Turner, M. (2006). Product design and manufacturing process improvement using association rules.Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 220(2), 243-254.

Shapiro, A. (2006). Stochastic programming with equilibrium constraints.Journal of Optimization Theory and Applications, 128(1), 221-243.

Suh, N. P. (2001). Axiomatic Design: Advances and Applications (The Oxford Series on Advanced Manufacturing).

Thevenot, H. J., & Simpson, T. W. (2007). A comprehensive metric for evaluating component commonality in a product family. Journal of Engineering Design, 18(6), 577-598.

Tonshoff, H. K., Beckendorff, U., & Andres, N. (1989, September). FLEXPLAN: A concept for intelligent process planning and scheduling. InProceedings of the CIRP international workshop (pp. 319-322).

Treleven, M., & Wacker, J. G. (1987). The sources, measurements, and managerial implications of process commonality. Journal of Operations Management, 7(1), 11-25.

Tseng, H. E., Chang, C. C., & Chang, S. H. (2005). Applying case-based reasoning for product configuration in mass customization environments.Expert Systems with Applications, 29(4), 913-925.

Tseng, M. M., Jiao, J., & Merchant, M. E. (1996). Design for mass customization. CIRP Annals-Manufacturing Technology, 45(1), 153-156.

Tsubone, H., Matsuura, H., & Satoh, S. (1994). Component part commonality and process flexibility effects on manufacturing performance.THE INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH, 32(10), 2479-2493.

Ulrich, K. (1994). Fundamentals of product modularity (pp. 219-231). Springer Netherlands.

Ulrich, K. (1995). The role of product architecture in the manufacturing firm.Research policy, 24(3), 419-440.

Van Veen, E. A., & Wortmann, J. C. (1992). Generative bill of material processing systems. Production Planning & Control, 3(3), 314-326.

Vandenbroucke, B., & Kruth, J. P. (2007). Selective laser melting of biocompatible metals for rapid manufacturing of medical parts. Rapid Prototyping Journal, 13(4), 196-203.

Venkat, K., & Wakeland, W. W. (2006). Using Simulation to Understand and Optimize a lean service Process. Simulation Series, 38(2), 242.

Vicente, L., Savard, G., & Júdice, J. (1994). Descent approaches for quadratic bilevel programming. Journal of Optimization Theory and Applications, 81(2), 379-399.

Von Stackelberg, H. (1952). The theory of the market economy. Oxford University Press.

Waller, B. (2004). Market responsive manufacturing for the automotive supply chain. Journal of Manufacturing Technology Management, 15(1), 10-19.

Wang, G., Jin, C., & Deng, X. (2008, September). Modeling and optimization on steel plate pick-up operation scheduling on stackyard of shipyard. InAutomation and Logistics, 2008. ICAL 2008. IEEE International Conference on(pp. 548-553). IEEE

Wang, R., & Dagli, C. H. (2008, April). An executable system architecture approach to discrete events system modeling using SysML in conjunction with colored Petri Net. In Systems Conference, 2008 2nd Annual IEEE (pp. 1-8). IEEE.

Wang, W. P. (2009). Toward developing agility evaluation of mass customization systems using 2-tuple linguistic computing. Expert Systems with Applications, 36(2), 3439-3447.

Wang, Y. F., Zhang, Y. F., Fuh, J. Y. H., Zhou, Z. D., Lou, P., & Xue, L. G. (2008, September). An integrated approach to reactive scheduling subject to machine breakdown. In Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on (pp. 542-547). IEEE.

Warmer, J. B., & Kleppe, A. G. (2003). The object constraint language: getting your models ready for MDA. Addison-Wesley Professional.

Weintraub, A., Cormier, D., Hodgson, T., King, R., WIlson, J., & Zozom, A. (1999). Scheduling with alternatives: a link between process planning and scheduling. Iie Transactions, 31(11), 1093-1102.

Williams, C. B., Allen, J. K., Rosen, D. W., & Mistree, F. (2007). Designing platforms for customizable products and processes in markets of non-uniform demand. Concurrent Engineering, 15(2), 201-216.

Wolter, J. D. (1991). On the automatic generation of assembly plans (pp. 263-288). Springer US.

Wortmann, H. C., & Erens, F. J. (1995, February). Control of variety by generic product modeling. In Proc. First World Congress on Intelligent Manufacturing Processes and Systems (Vol. 2, pp. 1327-1342).

Wu, C. T., & Egbelu, P. J. (1994, June). Concurrent design of shop layout and material handling. In Proceedings of the International Material Handling Research Colloquium (pp. 13-15).

Xue-hong, Y. U. (2009). Statistical Interconnect Timing Models Using Symbolic Moments. Microelectronics & Computer, 7, 018.

Yang, D., Jiao, J. R., Ji, Y., Du, G., Helo, P., & Valente, A. (2015). Joint optimization for coordinated configuration of product families and supply chains by a leader-follower Stackelberg game. European Journal of Operational Research, 246(1), 263-280.

Yang, D., Miao, R., Wu, H., & Zhou, Y. (2009). Product configuration knowledge modeling using ontology web language. Expert Systems with Applications, 36(3), 4399-4411.

Yang, T., Peters, B. A., & Tu, M. (2005). Layout design for flexible manufacturing systems considering single-loop directional flow patterns.European Journal of Operational Research, 164(2), 440-455.

Yeh, J. Y., & Wu, T. H. (2005). Solutions for product configuration management: An empirical study. AIE EDAM, 19(01), 39-47.

Zangiacomi, A., Zhijian, L., Sacco, M., & Boër, C. R. (2004). Process planning and scheduling for mass customised shoe manufacturing. International Journal of Computer Integrated Manufacturing, 17(7), 613-621.

Zha, X. F., Lim, S. Y. E., & Fok, S. C. (1998). Integrated intelligent design and assembly planning: a survey. The International Journal of Advanced Manufacturing Technology, 14(9), 664-685.

Zhang, T., & Efstathiou, J. (2006). The complexity of mass customization systems under different inventory strategies. International Journal of Computer Integrated Manufacturing, 19(5), 423-433.

Zhao, Y., & Fan, Y. S. (2007). Implementation approach of ERP with mass customization. International Journal of Computer Integrated Manufacturing,20(2-3), 160-168.

Zhu, X., Hu, S. J., Koren, Y., & Marin, S. P. (2008). Modeling of manufacturing complexity in mixed-model assembly lines. Journal of Manufacturing Science and Engineering, 130(5), 051013.

VITA

YITAO LIU

LIU was born in Shijiazhuang, Hebei Province, China. He obtained his Bachelor's degree in Mechanical Engineering and Automation from Beijing University of Chemical Technology, Beijing, China in 2008; and received his Master's degree in Mechanical Engineering from Beijing University of Aeronautics and Astronautics, Beijing, China in 2011. He is expected to obtain his Ph.D. degree in Mechanical Engineering at Georgia Institute of Technology, Atlanta, GA in 2016. When he is not working on his research, he enjoys photography, astronomy observation, playing badminton, and hiking with friends and family.

His main research interests include engineering design, manufacturing systems engineering, design optimization and automation. He has joint industrial collaboration projects on manufacturing systems engineering and automation with Renewable Bioproducts Institute (used to be called Institute of Paper Since and Technology) and Tyco Electronics Corporation for last four years. He has published 7 journal and conference papers and 6 invention patents during his research in Georgia Tech.