CO-EVOLUTION OF PRODUCT FAMILIES AND ASSEMBLY SYSTEMS

by

April M. Bryan

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mechanical Engineering)
in The University of Michigan
2008

Doctoral Committee:

Professor Shixin Jack Hu, Co-Chair
Professor Yoram Koren, Co-Chair
Professor Panos Y. Papalambros
Associate Professor Jionghua Jin
Jeffrey Abell, General Motors Corporation

To my parents

ACKNOWLEDGEMENTS

As I reflect on my studies, there are numerous people who have influenced my journey. I would now like to express my gratitude to them.

I thank Prof. Hu and Prof. Koren, my co-advisors. Through the numerous meetings we had together, I learned to defend my intellectual viewpoints. I also thank Prof. Hu for supporting me throughout my time as a student. Through working with him, I have become a stronger academic and person. I am immensely grateful to Prof. Koren for his continuous encouragement, without which I may not have realized this goal. Prof. Koren's influence extended beyond research, as he continuously advised me on the importance of developing leadership skills and giving back to the community.

I thank the other members of my thesis committee. I thank Prof. Papalambros for continuously giving me advice. I thank Jeff Abell from GM for securitizing my research and for providing my work with an industrial perspective. I thank Prof. Judy Jin for serving on my committee on such short notice.

I also like to thank my non-committee mentors Dr. Susan Montgomery, Dr. Reuven Katz, Prof. Levi Thompson, and Prof. Zbigniew Pasek for all their advice and support.

My research was funded first by the General Motors Resarch Laboratory for Advanced Vehicle Manufacturing (GMCRL), and then by the Engineering Research Centers for Reconfigurable Manufacturing Systems (ERC). I would like to thank the sponsors of these research programs for their support.

I would like to acknowledge my labmates. I thank Dr. Jianpeng Yue and Dr. Jaime Camelio for being my first research mentors. I also thank Oben Ceryan and John Wang for being such great labmates and for the numerous interesting discussions that we shared. I thank other members of the Hu-Lab and ERC with whom I had an opportunity to work.

I would like to thank the support staff at the GMCRL, the ERC, and the Office of Academic Services for their assistance, efficiency and friendship. I especially thank Tonya Marion, Rodney Hill, Lenea Howe, and Kathleen Brothchner. I would also like to thank Cynthia Quann-White for helping me to navigate the Ph.D. requirements and her general guidance

I have also been a part of numerous research groups and professional organizations that have been very helpful to me. I especially thank my colleagues in the Design Science Colloquium, Optimal Design Laboratory, The Society of Minority Engineers and Scientists Graduate Component, The Movement of Underrespresented Sisters in Engineering and Science, the American Society of Engineering Education, and Women in Science and Engineering.

I would also like to thank my family and friends for being a constant source of support and encouragement. Last, but not least, I thank God the numerous blessings that he has bestowed on my life.

TABLE OF CONTENTS

DEDICATION ……………………………………………………………...……..ii

ACKNOWLEDGEMENTS ………………………………………………….....iii

LIST OF FIGURES ……………………………………………..…..…...…..…..ix

LIST OF TABLES ……………………………………………...……………..….xi

CHAPTER

1. INTRODUCTION ……………………………………………...…………….1

1.1 Motivation ……………………………………………...……..…………1

1.2 Overview of the Co-Evolution Methodology ...………………………...…..4

1.3 Summary of Research Objectives ……...………...……………………...…....6

1.4 Organization of this Dissertation ...……………………………………...….7

References ……………………………………………………...…....……..10

2. CONCURRENT PRODUCT PORTFOLIO PLANNING AND MIXED

PRODUCT ASSEMBLY LINE BALANCING …………………………………...11

Abstract ……………………………………………………………………....11

2.1 Introduction ………………………………………………………...………..12

2.2 Definitions ……………………………………...…………………………15

2.3 Concurrent Product Portfolio Planning and Mixed Product Assembly Line

Balancing Problem ……………………………………...……………...17

2.3.1 Generation of the Initial Population ………………………………18

2.3.2 Sub-Problem 1: Computation of Oversupply …………………......19

2.3.3 Sub-Problem 2: Mixed Product Assembly Line Balancing …..…..20

2.4 Example of Methodology ………………………………...…………22

2.4.1 Validation of Methodology ...……………………………………...22

2.4.2 Product Portfolio Example ……………………………...……26

2.5 Conclusions ...……………………………………………………………..27

Acknowledgements ...………………………………………………………27

Nomenclature …….……………………………………………...………28

References ……………………………………………………………………30

3. CONCURRENT DESIGN OF PRODUCT FAMILIES AND ASSEMBLY SYSTEMS FOR PROFIT MAXIMIZATION ...…………………………………..32

Abstract ...…………………………………………………………………..32

3.1 Introduction ...…………………………………………………………..…33

3.2 Product Family Revenue and Cost Models ...………………………………..36

3.2.1 Product Family Representation ...…………………………………36

3.2.2 Product Family Revenue ...……………………………………...38

3.2.3 Production Cost for the Product Family ...………………………..41

3.3 Optimization Formulation ...…………………………………………………46

3.4 Solution Approach ...………………………………………………...48

3.4.1 Chromosome Representation, Encoding and Decoding ...………...48

3.4.2 Initial Population ...…………………………………………...50

3.4.3 Selection 51

3.4.4 Genetic Manipulation 51

3.5 Examples 54

3.5.1 Example 1 54

3.5.2 Larger Examples 59

3.5.3 Discussion 60

3.6 Conclusions 61

Acknowledgements 62

Nomenclature 62

Appendices 64

References 68

4. ASSEMBLY SYSTEM RECONFIGURATION PLANNING 70

Abstract 70

4.1 Introduction 70

4.2 Representations 73

4.2.1 Product Family Representation 73

4.2.2 Assembly System Representation 74

4.2.3 Assembly System Reconfiguration Planning 76

4.3 Mathematical Model 77

4.3.1 Life Cycle Costs 77

4.3.2 Optimization Formulation 82

4.4 Solution Approaches 84

4.4.1 Algorithm for Grouping Sequences into Subsets 85

4.4.2 Dynamic Programming 86

4.4.3 Genetic Algorithm 91

4.5 Examples 93

4.5.1 Example 1 94

4.5.2 Example 2 96

4.5.3 Discussion 99

4.6 Conclusions 100

Acknowledgements 101

Nomenclature 102

References 104

5. SUMMARY AND FUTURE RESEARCH 106

5.1 Summary 106

5.2 Original Contributions 109

5.3 Suggested Future Research 110

BIBLIOGRAPHY 114

LIST OF FIGURES

Figure 1-1 Comparison between concurrent engineering and the co-evolution of product families and assembly systems 4

Figure 1-2 Second phase of the co-evolution methodology 6

Figure 2-1 Product family architecture 13

Figure 2-2 Modules of a chair 23

Figure 2-3 Product family architecture of the chair 23

Figure 2-4 Precedence diagram for the chair...25

Figure 2-5 Line balancing solution for portfolio 1....................................25

Figure 2-6 Line balancing solution for portfolio 2....................................26

Figure 3-1 Features, modules, and product variant.....................................37

Figure 3-2 Precedence diagram for product variants and product family..............43

Figure 3-3 Chromosome representation..49

Figure 3-4 Decoding of the task sequence into workstations.............................50

Figure 3-5 Crossover operator..52

Figure 3-6 Mutation operator...53

Figure 3-7 Inversion operator..53

Figure 3-8 Office chair showing modules...55

Figure 3-9 Office chair precedence diagram..55

Figure 3-10 Workstation loads..58

Figure 3-11 Assembly line configurations...59

Figure 4-1 Assembly system representation……………………………………………74

Figure 4-2 Product life cycle…………………………………………………………...77

Figure 4-3 Representation of a sequence of items…………………………………...85

Figure 4-4 Algorithm for generating all the subsets of a sequence………………….86

Figure 4-5 Examples of assembly system layouts…………………………………...90

Figure 4-6 Examples of assembly system configurations…………………………...91

Figure 4-7 Chromosome representation……………………………………………..92

Figure 4-8 Precedence diagram for example 1………………………………………94

Figure 4-9 Optimal ASRP for example 1……………………………………………95

Figure 4-10 Net present costs for example 1…………………………………………….96

Figure 4-11 Precedence diagram for example 2………………………………………97

Figure 4-12 Net present life cycle cost …….………………………………………98

LIST OF TABLES

Table 2-1 Product variant matrix for the chair……………………………..24

Table 2-2 Assembly tasks for the chair ……...…………………………….24

Table 2-3 Production ratios for the portfolios....………………...………….25

Table 2-4 Demand rates for differentiating modules……………………….26

Table 3-1 Office chair modules and assembly times……………………….55

Table 3-2 Office chair parameters…………………………………………..56

Table 3-3 Results for the concurrent and the sequential approach………....57

Table 3-4 Product family assembly times for the office chair……………...57

Table 3-5 Genetic algorithm parameters for example 2 and example 3……60

Table 3-6 Results for example 2 and example 3……………………………60

Table 3A-1 Data of customer utility values for example 1…………………...64

Table 3B-1 The revenues for module instances for example 1……………….65

Table 3C-1 Parameters for examples 2 and 3………………………………...66

Table 3C-2 Example 2 modules and assembly times………………………...66

Table 3C-3 Example 3 modules and assembly times………………………...66

Table 3C-4 Precedence table for example 2………………………………….67

Table 3C-5 Precedence table for example 3………………………………….67

Table 4-1 Parameters for example 1………………………………………...95

Table 4-2 GA parameters for example 1……………………………………96

Table 4-3 GA parameters for example 2……………………………………97

Table 4-4 Example 2 –task-workstation assignments……………..………..98

Table 4-5 Example 2 with lower configuration costs - task-workstation assignments …………………………………………………...99

CHAPTER 1

INTRODUCTION

1.1 Motivation

When manufacturers had guaranteed markets for their products, they could remain profitable by producing a very limited number of variants of each product and introducing new products to market when they thought the product was sufficiently mature. Globalization and more empowered consumers presented new challenges for manufacturers. The globalization of markets resulted in the availability of a greater number of competitive products. At the same time, the increased purchasing power of consumers became the determining factor for the success of a product and the timing of the introduction of new products. In order to remain competitive, manufacturers increased the amount of variety that they supplied to the market and increased their responsiveness to changing market needs. Product families and reconfigurable manufacturing systems (RMS) are two enablers of high product variety and high responsiveness.

Product families have been defined as “sets of products that are derived from a common platform and yet possess specific features/functionality to meet particular customer requirements” [Meyer *et al*, 1997]. The product variants in product families typically have modular product architectures. It is through the selection of modules with differing characteristics that the differences in the functionality of the product variants of a product family are realized. The design of product families therefore involves the selection of modules for product variants, the selection of product variants for the product family, and the determination of the market share for each product variant. Consumer choice modeling techniques have been introduced for the design of product families [Li and Azarm, 2002, Green and Krieger, 1989]. This problem is known as the product line selection problem. Throughout this dissertation, the term product portfolio is used interchangeably with the term product family.

RMS was defined by Koren *et al* (1999) as follows: "A Reconfigurable Manufacturing System (RMS) is designed at the outset for rapid change in structure, as well as in hardware and software components, in order to quickly adjust production capacity and functionality within a part family in response to sudden changes in market or in regulatory requirements". More specifically, this dissertation is focused on the design of reconfigurable assembly systems. Whereas methods for the design of a single generation of an assembly system, and several generations of reconfigurable machining systems have been proposed, there is less research on the design of reconfigurable assembly systems. Mixed model assembly line balancing techniques are used in the design of an assembly system for a single generation of a product family [Becker and Scholl, 2006]. Reconfigurable machining systems are assumed to possess modular components. The design of reconfigurable machining systems involves the determination of how the system can be scaled up or converted as product volumes and functionality changes [Spicer, 2007, Son, 2000].

In a responsive and high variety manufacturing environment, strategies are also required for cost effectively evolving product families and their assembly systems. Some researchers have previously recognized a need for product evolution and assembly system reconfiguration [e.g. Frei *et al*, 2007, Seepersad *et al*, 2005, Martin and Ishii, 2002, Travaini *et al*, 2002]. However, methodologies for product evolution and assembly system reconfiguration have been pursued independently. There is a lack of systematic methods for concurrently evolving product families and assembly systems.

Co-evolution of product families and assembly systems is introduced in this dissertation as a new product development methodology for the joint design and reconfiguration of product families and assembly systems within and across product generations. The co-evolution method incorporates product variant and assembly system changes within a product family generation, as well as between generations through continuously reconfiguring product families and assembly systems. Co-evolution can enable manufacturers to remain competitive as it maximizes the reuse of product modules and reconfigurable systems to ensure that assembly systems are effective for as many product generations as possible [Bryan *et al*, 2007].

The methods for the co-evolution of product families and assembly systems that are introduced in this dissertation are different from earlier concurrent engineering processes as concurrent engineering plans for the present generation while co-evolution plans for both the present generation and future changes. With concurrent engineering techniques, usually each generation of a product family is associated with a unique assembly system. Through the concurrent planning of the product family and assembly system with co-evolution, essentially the same assembly system may be utilized for several product generations. Figure 1-1 illustrates the main differences between co-evolution and traditional concurrent engineering strategies.

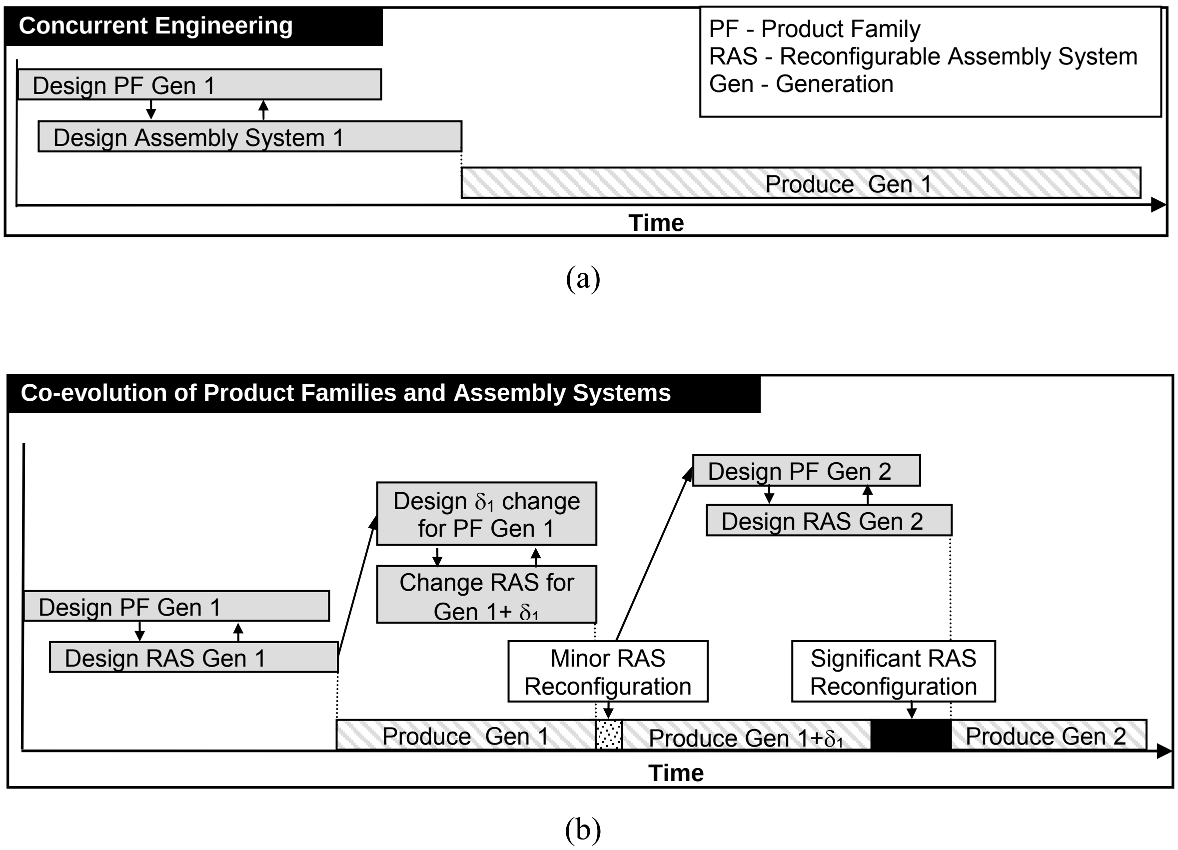
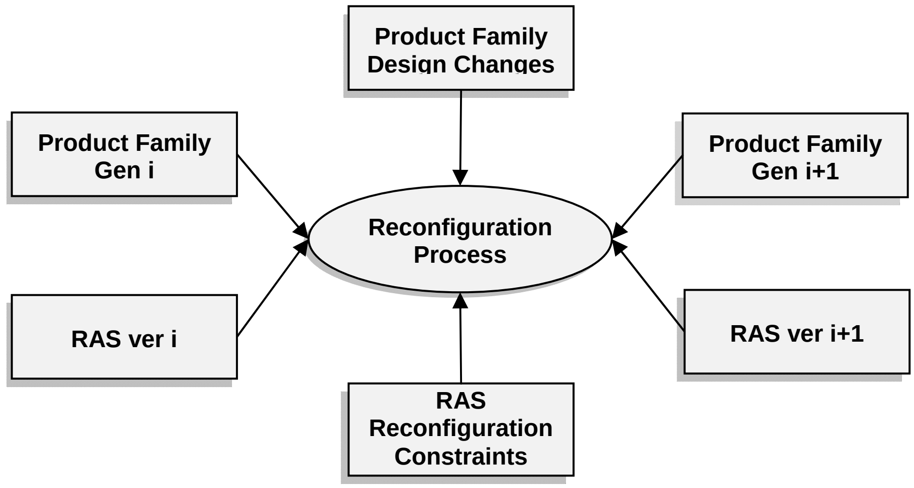


Figure 1-1 Comparison between concurrent engineering and the co-evolution of product families and assembly systems

1.2 Overview of the Co-Evolution Methodology

Co-evolution of product families and assembly systems is formally defined as a method for the joint design of and reconfiguration of the functionality and capacity of product families and their corresponding assembly systems within and across product generations in order to meet the present and future needs of the market. In most cases, the future needs of the market are not completely uncertain. The co-evolution method utilizes the information that is known about the future, such as the desired product functionality and projected market share, to make future plans. If the future needs are as predicted, the advanced planning of the co-evolution methodology allows for faster and more cost effective realization of new product families and assembly systems than if the planning was not performed. However, if the planned functionality and capacity are not needed, there may be some loss in initial investment. Throughout this dissertation, examples are used to illustrate the advantage of this advanced planning over cases when planning for future evolution is not utilized.

There are two main phases of the co-evolution methodology: (1) the joint design of the product family and assembly system in the first generation and (2) the later co-evolution of the product family and assembly system (Figure 1-2). The first generation design is assumed to be a ‘clean sheet’ design for both the product family and the assembly system. This initial design phase is important as the decisions made during this phase affect product family changes and assembly system reconfigurations in future generations. The co-evolution phase pursues economical product family changes and assembly system reconfigurations. As shown in Figure 1-2, the inputs to this phase are the existing product family, the required design changes, and the re-configuration constraints of the product family and assembly system. Using these inputs, new product modules are developed, the product family is modified, and the assembly system is reconfigured.



RAS –Reconfigurable Assembly

Figure 1-2 Second phase of the co-evolution methodology

1.3 Summary of Research Objectives

Implementation of the co-evolution methodology requires three main mathematical models:

(1) A model for the joint design of the product family and assembly system in the first generation.

(2) A model for the of the evolution of the product family within the constraints of the existing assembly system.

(3) A model for the reconfiguration of the assembly system in response to product family changes.

In this dissertation, I have developed the mathematical models and solution procedures for the first and last problems. The specific research tasks are as follows:

(1) Develop formulations and solution procedures for the concurrent design of the product family and assembly system in the first generation.

(2) Develop a method for reconfiguring the assembly system in order to produce an evolving product family.

1.4 Organization of this Dissertation

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

In Chapter 2, I introduce a formulation for the concurrent design of a product portfolio and an assembly system. This method approaches the problem of the concurrent design of the product family and assembly system from a purely operations perspective. A new integer program is developed for the selection of modules so that the oversupply of functionality is minimized. A traditional mixed-model, assembly line balancing problem is used to design an assembly system that maximizes efficiency while minimizing the variation between workstations and within workstations. These two objectives are combined into a single objective optimization problem in order to obtain the explicit design of the product family and the assembly system. A genetic algorithm is developed for solving this problem. The results obtained from implementation of this methodology indicate that the design of the product family does have an impact on the assembly system.

Whereas it is important to operate efficiently, the ultimate objective of manufacturers is profitability. Manufacturers can be profitable by providing products that are desirable to the market at the lowest possible cost. In chapter 3, the problem of the concurrent design of the product family and assembly system is reformulated as a profit maximization problem. The product design portion of the problem is based on the product line selection problem while assembly system design is based on mixed-model assembly line balancing techniques. By the consideration of market needs, the problem introduced in this chapter expands on the purely operations research problem introduced in chapter 2. A genetic algorithm is introduced for the solution of this problem. The output of this model includes the product variants for the product family, their market share, the assignment of tasks to workstations, and the number of workstations in the assembly system. An analysis of the results obtained from this problem indicates that the profit maximization problem for the concurrent design of product families and assembly systems leads to results as good as or better than the traditional sequential approach. In addition, the results show that whereas increasing the number of product variants in a product family may increase market share and revenues, it can also increase cost.

Chapter 4 presents the assembly system reconfiguration planning problem (ASRP) for selecting several generations of assembly systems that minimizes the life cycle cost of producing a product family. A new formulation for the life cycle cost of the product family is introduced in this chapter. The life cycle cost takes into consideration both the variable costs of producing the product family within each generation and the costs associated with reconfiguring the product family between product generations. Dynamic programming and genetic algorithm procedures are developed for solving the optimization problem. A new algorithm for generating all the possible configurations of a parallel-serial assembly system is introduced to generate all the states for the dynamic program. Examples are used to illustrate that by planning for future changes, the ASRP approach leads to lower total costs over several product generations than existing methods.

Chapter 5 summarizes the work presented and highlights the major contributions of this dissertation. Suggested future directions for research are also given in this chapter.

References

Becker, C., and Scholl, A., 2006, "A Survey on Problems and Methods in Generalized Assembly Line Balancing," *European Journal of Operational Research*, 168(3), pp. 694-715.

Bryan, A., Ko, J., Hu, S. J., and Koren, Y., 2007, "Co-Evolution of Product Families and Assembly Systems," *CIRP Annals*, 56(1), pp. 21-44.

Frei, R., Ribeiro, L., and Barata, J., 2007, "Evolvable Assembly Systems: Towards User Friendly Manufacturing," Proceedings of the IEEE International Symposium on Assembly and Manufacturing, Ann Arbor, Michigan, USA.

Green, P., and Krieger, A., 1989, "Recent Contributions to Optimal Product Positioning and Buyer Segmentation," *European Journal of Operations Research*, 41(2), pp.127-141.

Koren, Y., Heisel, U., Jovan, F., Moriwaki, T., Pritschow, G., Ulsoy, G., and Van Brussel, H., 1999, "Reconfigurable Manufacturing Systems," *CIRP Annals*, 48(2), pp. 527-540.

Li, H., and Azarm, S., 2002, "An Approach for Product Line Design Selection Under Uncertainty and Competition," *ASME Journal of Mechanical Design*, 124(3), pp.385-392.

Martin, M. V., and Ishii, K., 2002, "Design for Variety: Developing Standardized and Modularized Product Platform Architectures," *Research in Engineering Design*, 13(4), pp. 213-235.

Meyer, M. H., Tertzakian, P., and Utterback, J. M., 1997, "Metrics for Managing Research and Development in the Context of the Product Family," *Management Science*, 43(1), pp. 88-111.

Seepersad, C. C., Mistree, F., and Allen, J. K, 2005, "Designing Evolving Families of Products Using the Utility-Based Compromise Decision Support Problem," *International Journal of Mass Customisation*, 1(1), pp. 37-64.

Son, S. Y., 2000, *Design Principles and Methodologies for Reconfigurable Machining Systems*, Ph.D. thesis, University of Michigan, Ann Arbor, MI.

Spicer, P., and Carlo, H., 2007, "Integrating Reconfiguration Cost Into the Design of Multi-Period Scalable Manufacturing Systems," *ASME Journal of Manufacturing Science and Engineering*, 129(1), pp. 202-210.

Travaini, E., Pedrazzoli, O., Rinaldi, R., and Boor, C. R., 2002, "Methodological Approach and Reconfiguration Tool for Assembly Systems," *CIRP Annals*, 51(1) pp. 9-13

CHAPTER 2

CONCURRENT PRODUCT PORTFOLIO PLANNING AND MIXED PRODUCT ASSEMBLY LINE BALANCING

Abstract

Product families and reconfigurable manufacturing systems have enabled manufacturers to provide "cost effective" variety to the market. In spite of these advances, the high cost of manufacturing sometimes makes it infeasible for manufacturers to supply all the possible variants of a product to the market. Therefore, the determination of the right number of product variants to offer in product portfolios becomes an important consideration. The product portfolio planning problem had been independently well studied from marketing and engineering perspectives. However, advantages can be gained from using a concurrent marketing and engineering approach. Concurrent product development strategies for product families and reconfigurable manufacturing systems can allow manufacturers to select the best product portfolios from marketing, product design, and manufacturing perspectives.

This chapter presents a methodology for the concurrent design of a product

[1]A version of this chapter has been published. Bryan, A., Hu, S. J., and Koren, Y, 2007, "Concurrent Product Portfolio Planning and Mixed Product Assembly Line Balancing," Chinese Journal of Mechanical Engineering, 20(1), pp. 96-99.

portfolio and its corresponding assembly system. The objective of the concurrent product portfolio planning and assembly system design problem is to obtain the product variants that will make up the product portfolio such that the oversupply of differentiating modules is minimized and the efficiency of the assembly line is maximized. Explicit design of the assembly system is obtained in solving this problem. It is assumed that the demand for differentiating modules and the assembly times for these modules are known a priori. A genetic algorithm is used for solving this problem. The basic premise of this methodology is that the selected product portfolio has a significant impact on the solution of the assembly line balancing problem. An example is used to validate this claim. The example is then further developed to demonstrate how the methodology can be used to obtain the optimal product portfolio. This approach is intended for use by manufacturers during the early design stages of a product family.

2.1 Introduction

Global competition has led manufacturers to seek new strategies for increasing market share. One strategy often utilized by manufacturers is to increase the supply of product variety to the market with the intention that almost every customer may find a product that meets his/her needs. Product families and reconfigurable manufacturing systems (RMS) have provided manufacturers with the means to cost effectively supply this product variety to the market [Koren *et al*, 1999].

The conventional wisdom is that there is a direct correlation between an increase in product variety and an increase in a company's profits. However, an increase in product variety can sometimes result in diminishing returns as an increase in the number of product variants can lead to an increase in manufacturing costs [Child *et al*, 1991]. Therefore, some manufacturers opt to select specific product variants to form the product portfolio that is offered to the market. The manufacturers that use this strategy are then faced with the challenge of determining the right number of product variants to offer in the company's product portfolio.

The solution to the product portfolio problem was first approached from a purely marketing perspective. The marketing approach considers the product as a set of features or attributes from which the product variants are derived. Tools such as utility modeling and conjoint analysis are used to determine the product variants that are necessary to meet the market's needs [e.g. Krishnan and Ulrich, 2001, Kaul and Rao, 1995, Green and Srinivasan, 1990, Green and Krieger, 1987].

More recently, the product portfolio planning problem has been approached from an engineering perspective. In the engineering approach, the product is viewed as complex assemblies of interacting components or modules, rather than as a set of product functionalities [Fellini *et al*, 2005, Michalek *et al*, 2005]. The selection of components and component attributes are based on the product's technical performance, innovativeness, and effectiveness [Michalek *et al*, 2005].

The determination of the product portfolio from either the purely marketing or engineering approach can lead to the selection of sub-optimal product portfolios. Concurrent engineering strategies for product families allow manufacturers to determine the right amount of "cost effective" variety to offer in order to remain competitive. There is limited research in this field [Jiao and Zhang, 2005, Michalek *et al*, 2005, Abdi and Labib, 2004]. Jiao and Zhang (2005) extended the conjoint analysis formulation for the product portfolio planning problem to include the consideration of the process capability index which gives an indication of the degree of customizability of the process. As the process capability index increases, the ability to manufacture a wide range of product variants on the assembly system increases. Abdi and Labib (2004) used the analytical hierarchical process (AHP) to select the product variants to be produced so that there was a high degree of process similarity between the components of the product family. Michalek *et al* (2005) used analytical target cascading (ATC) for the maximization of the revenue from the product line and the minimization of the cost of the manufacturing system.

Although the previous strategies considered measures for manufacturability during the product design process, none of them led to the explicit design of the manufacturing system. In fact, all the previous approaches assume that the manufacturing system was designed a priori. There remains a need for not only the determination of the impact of a known manufacturing system on the design of a product portfolio, but for a method that allows for the explicit design of the manufacturing system during the product portfolio planning stage. Such a strategy could lead to further reductions in product development time and costs.

This chapter presents a methodology for the concurrent determination of the product portfolio and its corresponding mixed product assembly line. The method assumes that there is a one-to-one mapping between product functionalities and the modules or components that provide these functionalities. It also assumes that the demands for various product functionalities are known and satisfied. A multi-objective optimization formulation that minimizes the oversupply of differentiating modules and maximizes the efficiency of the assembly line is developed. The concurrent product portfolio planning and mixed product assembly line balancing problem is solved using a genetic algorithm. The members of the population are the product portfolios and the genes are the product variants. The fitness value of each member of the population is determined in three phases. The first stage involves the determination of the production ratios for the product variants in the product portfolio. The next stage involves finding a solution to the mixed product assembly line balancing problem for the product portfolio using the formulation developed by Rekiek *et al* (2000). The overall fitness value is computed in the third stage.

This model is based on the assumption that the selected product portfolio does have an impact on the assembly line balancing solution. An example is used to validate this assumption and to demonstrate the implementation of the proposed method.

2.2 Definitions

This section reviews the terminology for product portfolio planning. The symbols used are explained in the Nomenclature section at the end of this chapter.

The decomposition tree of a product into its modules and instances is known as the product family architecture (Fig. 2-1). The product family architecture has been defined as "… a firm's product platform, within which various product variants can be derived from basic product designs to satisfy a spectrum of customer needs related to various market niches" [Jiao and Tseng, 2000]. It is assumed that the products in the product portfolio have completely modular architectures.

A *functional requirement (FR)* is defined as the feature that a customer desires to have in a product. Functional requirements can have one or more levels. There can be a level to not have the functional requirement at all. A *module/component* is the basic unit from which a product variant is derived. In the remainder of this chapter, the term module is used to refer to both modules and components. Since the product architecture is modular, each module fulfills exactly one functional requirement at one level. *Instances of a module* refer to modules with different levels of the same functional requirement. There is a *null instance* that corresponds to the level not to have a functional requirement. There are two types of modules, *base modules* and *differentiating modules*. Base modules are standard modules with only one instance and differentiating modules are modules with more than one instance. While customers have no choice in the level of a functional requirement provided by a base module, they can select the level of a functional requirement provided by a differentiating module by selecting the instance of the differentiating module that provides the functional feature at the desired level.

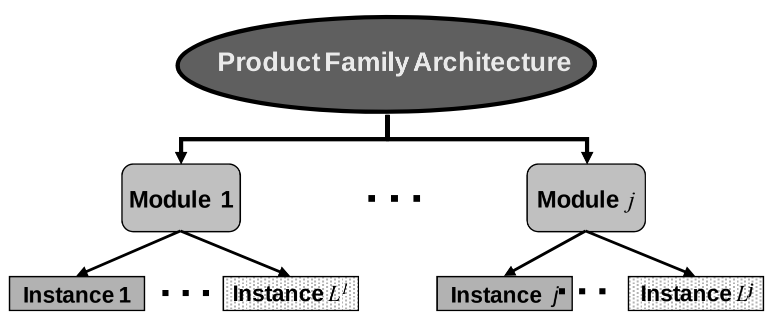


Figure 2-1 Product family architecture

A product variant is an individual product that consists of exactly one instance of each module. Considering S as the set of all product variants in the product family: S={v_i:

$v_1, v_2, \ldots, v_V\}$ where v_i is a product variant and the subscript V is the total number of product variants in the product family. A product portfolio (P) is defined as the subset of product variants, i.e. $P \subseteq S$ that is selected for production.

2.3 Concurrent Product Portfolio Planning and Mixed Product Assembly Line Balancing Problem

The concurrent product portfolio planning and mixed product assembly line balancing methodology determines the mix and combination of product variants to include in the product portfolio so that customer demand is satisfied, the oversupply of functionality is minimized, and the efficiency of the assembly line is maximized. The decisions include the product variants to include in the product portfolio, v_i, the production rates of the product variants, p_i, the assignment of modules to workstations, x_{jk}, and the number of workstations required for assembly, n. The objective function for this problem is defined as follows:

$$Maximize\ F_{tot} = \frac{1}{1+\Theta} + \varepsilon \qquad (2\text{-}1)$$

where F_{tot} is the objective function, Θ is the total oversupply of functional features in the product portfolio, and ε is a measure of the efficiency of the assembly line. Formulations for Θ and ε are developed in Sects. 2.3.2 and 2.3.3 respectively.

The optimization problem is solved using a genetic algorithm. Each chromosome is a potential product portfolio and the genes are the product variants. The chromosomes have variable lengths which are dependent on the number of product variants that are selected for the product portfolio. The initial population is randomly generated. The fitness value of each member of the population is determined in three stages. The production rates for product variants and the oversupply of functional features are computed in the first stage. The next stage involves the determination of the number of workstations, the assignment of tasks to workstations, and the line balancing efficiency for the selected portfolio. The oversupply of functional features and the efficiency of the assembly line from these two sub-problems are then used to calculate the fitness function.

2.3.1 Generation of the Initial Population

The possible product variants are generated using combinations of modules. The modules are assigned to product variants such that there is only one instance for each module in a product variant. For this assumption to hold, the conditions in Eq. (2-2 a) and Eq. (2-2 b) must be true.

$$\sum_{\ell=1}^{L^j} B_{ij\ell} = 1 \tag{2-2a}$$

$$\sum_{i=1}^{V} B_{ij\ell} \le V \tag{2-2b}$$

All the product variants are stored in a V x J product variant matrix in which the rows represent the product variants and the columns represent the modules. The value of each cell represents the instance of the module in the given product variant. It can be observed that selection of a product variant, a row of the matrix, automatically results in the selection of instances for the modules. Once all the product variants are generated, subsets of the product variants are randomly selected to form candidate product portfolios such that $P \subseteq S$. These subsets are stored as chromosomes in the initial population.

2.3.2 Sub-Problem 1: Computation of Oversupply

Using $D_{j\ell}$ as the known demand for instance ℓ of module j, and p_i as the production rate for each product variant, the linear program in Eq. (2-3) is used to determine the values of the variables, p_i. The inequality in Eq. (2-3) ensures that the minimum demand for product variants is satisfied. The equality guarantees the feasibility of the product portfolio by ensuring that the production rates of the product variants (p_i) do not exceed the market size.

$$\begin{aligned}
&\sum_{i=1}^{V} B_{ij\ell} p_i \geq D_{j\ell} \ \forall \ j = 1, \cdots, J; \ell = 1, \cdots, L^j \\
&\sum_{i=1}^{V} p_i \leq 1 \\
&0 \leq p_i \leq 1
\end{aligned} \tag{2-3}$$

Once the decision variables for the linear program, p_i, have been determined, Eq. (2-4) is used to determine Θ, the oversupply for the product portfolio.

$$\Theta = \sum_{j=1}^{J} \sum_{\ell=1}^{L^J} \left| \Phi_{j\ell} \right| \tag{2-4}$$

$\Phi_{j\ell}$ is defined as indicated in Eq. (2-5).

$$\Phi_{j\ell} = D_{j\ell} - q_{j\ell} \tag{2-5}$$

where,

$$q_{j\ell} = \sum_{i=1}^{V} B_{ij\ell} p_i \tag{2-6}$$

The selected product portfolio is used for assembly line design as described in the following section.

2.3.3 Sub-Problem 2: Mixed Product Assembly Line Balancing

The mixed product assembly line balancing sub-problem is used to determine the minimum number of workstations required to assemble the selected product portfolio and to maximize the efficiency of the assembly line. It has been noted that the cost of an assembly line is directly related to its efficiency [Becker and Scholl, 2006]. Therefore, by maximizing the efficiency of the assembly line, the cost of the assembly line can be minimized.

In the first stage of this approach, the general precedence diagram for the product is converted into a precedence diagram for delayed product differentiation. The product family precedence diagram is obtained from the precedence diagrams for delayed product differentiation of the individual product variants. The task time for a module instance, $T_{j\ell}$, is a given parameter. Equation (2-7) gives the computation of $TV_{ij\ell}$, the task time for a module instance in a product variant, and Eq. (2-8) gives the computation of the weighted average assembly task time for modules in the product family precedence diagram, TP_j. The product family precedence diagram is then used for the task assignment.

$$TV_{ij\ell} = B_{ij\ell} T_{j\ell} \tag{2-7}$$

$$TP_j = \sum_{i=1}^{V} \sum_{\ell=1}^{L^j} B_{ij\ell}\, p_i T_{j\ell} \;\; \forall\; j = 1, \cdots, J \tag{2-8}$$

Considering C as the assigned cycle time for assembly, the objective function and constraints for the mixed product assembly line balancing problem are stated in Eq. (2-9). n, the number of workstations required for assembly, and x_{jk}, the assignment of module j to workstation k are decision variables of the problem.

$$\begin{aligned} \mathit{Maximize}\;\; \varepsilon \;=\; & \frac{1}{3nC} \sum_{k=1}^{n} \sum_{j=1}^{J} x_{jk} TP_j \\ & + \frac{1}{3\left\{1 + \frac{1}{n}\left[\frac{\left|\sum_{j=1}^{J} x_{jk} TP_j - \frac{1}{n}\sum_{k=1}^{n}\sum_{j=1}^{J} x_{jk} TP_j\right|}{\frac{1}{n}\sum_{k=1}^{n}\sum_{j=1}^{J} x_{jk} TP_j}\right]\right\}} \\ & + \frac{1}{3\left\{1 + \frac{1}{n}\frac{1}{V}\sum_{k=1}^{n}\sum_{i=1}^{V}\left[\frac{\left|\sum_{j=1}^{J}\sum_{\ell=1}^{L^j} x_{jk} TV_{ij\ell} - \frac{1}{V}\sum_{i=1}^{V}\;\sum_{j=1}^{J}\sum_{\ell=1}^{L^j} x_{jk} TV_{ij\ell}\right|}{\frac{1}{V}\sum_{i=1}^{V}\;\sum_{j=1}^{J}\sum_{\ell=1}^{L^j} x_{jk} TV_{ij\ell}}\right]\right\}} \end{aligned} \tag{2-9}$$

subject to:

$$\sum_{k=1}^{n} x_{jk} = 1 \quad \forall\, j = 1, \cdots, J$$

$$x_{jk} \le \sum_{h=1}^{k} x_{gh} \;\; \forall\, j = 1, \cdots, J;\;\; k = 1, \cdots, n;\;\; \mathit{and}\;\; g \in Q(j)$$

$$\sum_{j=1}^{J} x_{jk} TP_j \le C \;\; \forall\;\; k = 1, \cdots, n$$

$$x_{jk} \in \{0,1\};\, n \ge 0$$

The first constraint in Eq. (2-9) ensures that each task is assigned to exactly one workstation. The second constraint prevents violation of the precedence constraints. *Q(j)* in this latter constraint represents the set of all modules that must be assembled prior to module j. The third constraint ensures that the cycle time for assembly is not exceeded at any workstation.

The mixed product assembly line balancing problem is solved using the methodology presented by Rekiek *et al* (2000). Using the assumption that the assembly line has a serial configuration, a grouping genetic algorithm is used to solve the optimization problem. The initial population is generated by using the bin packing approach [Garey and Johnson 1979]. The workstation solution obtained from the bin packing problem forms the chromosomes. The individual workstations are the genes, and the tasks that are assigned to the workstations are alleles. An elitist strategy is used for selection.

2.4 Example of Methodology

2.4.1 Validation of Methodology

In this section, it will be shown that the product portfolio does have an impact on the solution of the mixed product assembly line balancing problem.

Consider the case of the assembly of a chair that has the modules, and the product family architecture are shown in Figs. 2-2 and 2-3. The chair consists of nine modules, six base modules and three differentiating modules, therefore j=1,2,…,9. Modules 2,5,9 are the differentiating modules. Each of the differentiating modules has two instances:

L^j=2 for j =2,5,9. One of these instances is the null instance to not have the module at all. The cycle time for production is 16 sec.

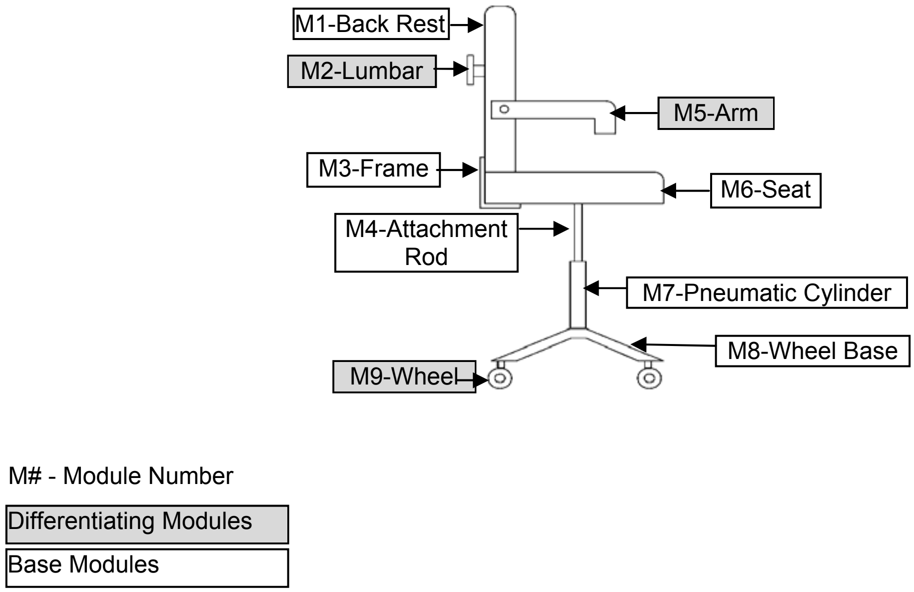


Figure 2-2 Modules of a chair

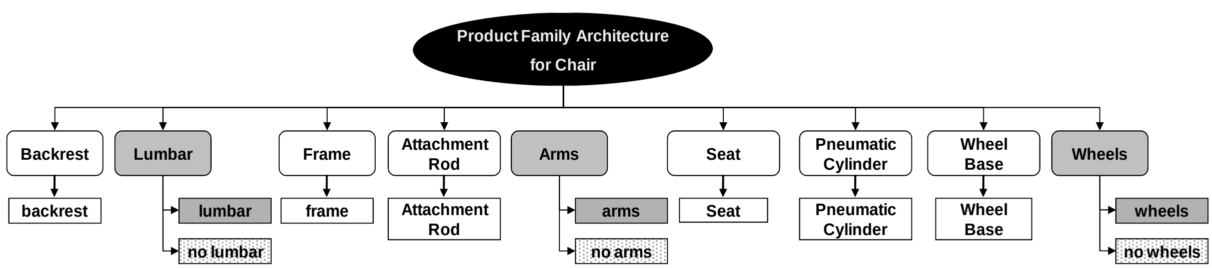


Figure 2-3 Product family architecture of the chair

Table 2-1 is the product variant matrix that shows the various combinations of module instances that form the product family. For clarity, the prefixes M and V are used to identify the modules and variants respectively. As shown in the table, there are eight possible product variants for this product family. The assembly tasks and the corresponding task times are given in Table 2-2. Figure 2-4 is the precedence diagram for the assembly of these tasks.

Table 2-1 Product variant matrix for the chair

Variants	Modules		
V1	2	2	2
V2	1	2	2
V3	2	1	2
V4	2	2	1
V5	1	1	2
V6	1	2	1
V7	2	1	1
V8	1	1	1

Table 2-2 Assembly tasks for the chair

Assembly Task	A	B	C	D	E	F	G	H
Task Description	Pneumatic Cylinder + Attachment Rod	+ Wheel Base	+ Wheels	+Frame	+Seat	+ Arms	+ Lumbar	+ Back Rest
Assembly Time for First Instance (sec)	3	5	4	8	7	5	10	8
Assembly Time for Second Instance (sec)	-	-	0	-	-	0	0	-

It is assumed that there is a case which results in two product portfolios with production rates as defined in Table 2-3. Solutions for the assembly line balancing problem for this case are shown in Figures 2-5 and 2-6. It is observed that the different product portfolios have different assembly line balancing solutions. This result is due to the fact that the production rates for the product variants in the two product portfolios are different. Therefore, the product family task times (Eq. (2-8)), which are used for task assignment, are different. These results justify the use of the methodology presented in

Sect. 2.3 for the concurrent product portfolio and mixed product assembly line design problem.

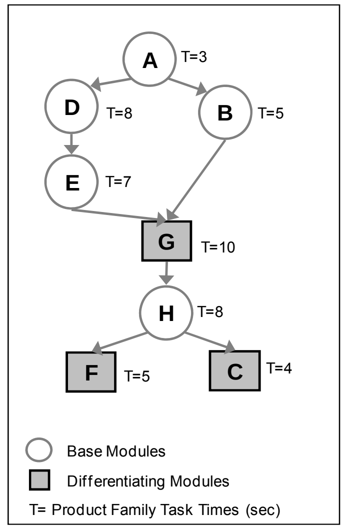


Figure 2-4 Precedence diagram for the chair

Table 2-3 Production rates for the portfolios

	Variants							
	V1	***V2***	***V3***	***V4***	***V5***	***V6***	***V7***	***V8***
Portfolio 1	0	0	0	0	0	0	0.875	0.125
Portfolio 2	0	0	0.5	0	0.075	0.3	0	0.125

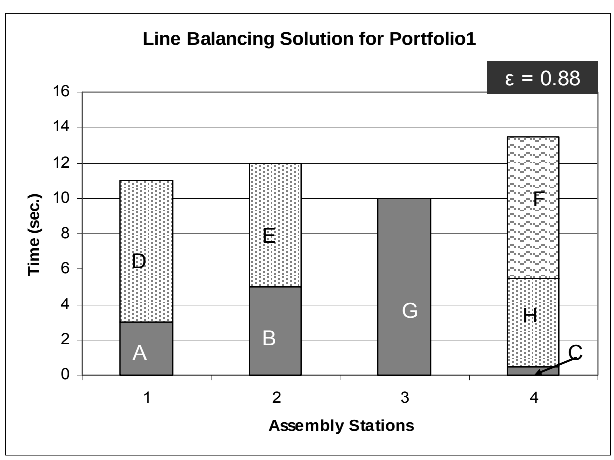


Figure 2-5 Line balancing solution for portfolio 1

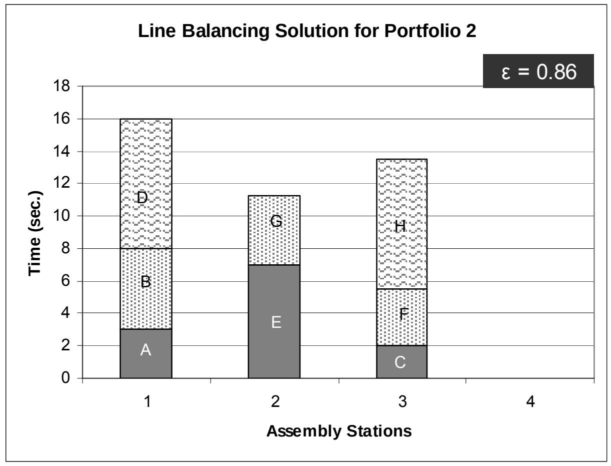


Figure 2-6 Line balancing solution for portfolio 2

2.4.2 Product Portfolio Example

The details for the current analysis of the chair are the same as discussed in the previous section. The product demands for the instances of the differentiating modules are provided in Table 2-4. One eighth of the customers do not want any of the differentiating modules.

Table 2-4 Demand rates for differentiating modules

	M2	M5	M9
Instance1	0.5	0.8	0.2
Instance2	0.5	0.2	0.8

The results indicate that to maximize the objective function, only the base product variant (V1) and the product variant that includes all the differentiating modules (V8) should be included in the product portfolio. The production rates of product variants V1 and V8 are 12.5% and 80% respectively. The value of the objective is F_{tot}=1.5 (Θ=0.9;ε=0.98). The assignment of tasks for this problem is the same as shown Fig. 2-6.

2.5 Conclusions

Concurrent product development strategies for product families and manufacturing systems can lead to an increase in responsiveness for manufacturers. In this chapter, a methodology for concurrent product portfolio planning and mixed product assembly line balancing is presented. The method assumes that the manufacturer satisfies customer demand for differentiating modules. The objective function of the problem presented minimizes this oversupply of differentiating modules and maximizes the efficiency of the assembly line. The main outputs of the formulation is the product variants in the portfolio for a given customer demand, the production rates of the selected product variants, the number of workstations required in the assembly system, and the assignment of tasks to workstations. It is observed that the presented approach finds the optimal portfolio as well as the explicit design of the assembly system. This is one of the main advantages of this methodology. This approach is most useful during the early stages of product family design. The results of the approach are dependent on the values for customer demand and the precedence diagram for the product family. Therefore as the design evolves, the method should be repeated.

Acknowledgements

The author would like to acknowledge co-authorship with Prof. S. Jack Hu and Prof. Yoram Koren. The author would also like to acknowledge the support of the Engineering Research Center for Reconfigurable Manufacturing Systems at the University of Michigan (NSF Grant No. EEC95-92125).

Nomenclature

Indices

i = product variant

j = module

k = workstation

ℓ = instance of module

Variables

n = number of workstations

p_i = production rate of product variant i

$q_{j\ell}$ = production rate of instance ℓ of module j (intermediate variable)

v_i = product variant i

x_{jk} = indicator assignment variable. 1 if module j is assigned to be assembled at workstation k; 0 otherwise.

Functions and Parameters

$B_{ij\ell}$ = parameter indicating that instance ℓ of module j is in product variant I 1 if the previous statement is true; 0 otherwise.

C = cycle time

$D_{j\ell}$ = demand for instance ℓ of module j

F_{tot} = objective function

J = maximum number of modules

L^j = maximum number of instances of module j

P = subset of product variants from set S selected for the product portfolio

$Q(j)$ = set of modules that must be assembled before module j

S = set of all product variants in the product family

$T_{j\ell}$ = time required for assembly of instance ℓ of module j

TP_j = weighted average assembly time for module j in the product family

$TV_{ij\ell}$ = time required for assembly of instance ℓ of module j in product variant i

V = maximum number of product variants

Greek Letters

ε = efficiency of the assembly system

Θ = oversupply of module instances in the product portfolio

$\Phi_{j\ell}$ = oversupply of instance ℓ of module j

References

Abdi, M. R., and Labib, A. W., 2004, "Grouping and Selecting Products: The Design Key of Reconfigurable Manufacturing Systems," *International Journal of Production Research*, 42(3), pp. 521-546.

Becker, C., and Scholl, A., 2006, "A Survey on Problems and Methods in Generalized Assembly Line Balancing," *European Journal of Operational Research*, 168(3), pp. 694-715.

Child, P., Diederichs, R., and Sanders, F. H., 1991, "The Management of Complexity," *Sloan Management Review*, 33(1), pp. 73-80.

Fellini, R., Kokkolaras, M., Papalambros, P., and Perez-Duarte, A., 2005, "Platform Selection Under Performance Bounds in Optimal Design of Product Families," *ASME Journal of Mechanical Design*, 37(4), pp. 524-535.

Garey, M. R., and Johnson, D. S., 1979, *Computers and Intractability: A Guide to NP-Completeness*, W. H. Freeman, San Francisco, California, USA.

Green, P. E., and Krieger, A. M., 1987, "Consumer Based Approach to Designing Product Line Extensions," *Journal of Product Innovation Management*, 4(1), pp. 21-32.

Green, P. E., and Srinivasan, V., 1990, "Conjoint Analysis in Marketing: New Developments with Implications for Research and Practice," *Journal in Marketing*, 54(4), pp. 3-19.

Jiao, J., and Tseng, M. M., 2000, "Fundamentals of Product Family Architecture," *Integrated Manufacturing Systems*, 11(7), pp. 469-483.

Jiao, J., and Zhang, Y, 2005, "Product Portfolio Planning with Customer-Engineering Interactions," *IIE Transactions*, 37(9), pp. 801-814.

Kaul, A., and Rao, V. R., 1995, "Research For Product Positioning and Design Decisions: An Integrative Review," *Research in Marketing*, 12(4), pp. 293-320.

Koren, Y., Heisel, U., Jovan, F., Moriwaki, T., Pritschow, G., Ulsoy, G., and Van Brussel, H., 1999, "Reconfigurable Manufacturing Systems," *CIRP Annals*, 48(2), pp. 527-540.

Krishnan, V., and Ulrich, K. T., 2001, "Product Development Decisions: A Review of the Literature," *Management Science*, 47(1), pp. 1-21.

Michalek, J. J., Ceryan, O., Papalambros, P., and Koren, Y., 2005, "Manufacturing Investment and Allocation in Product Line Design Decision-Making," Proceedings of the ASME Design Engineering Technical Conference, DETC2005-84812, Long Beach, California, USA.

Rekiek, B., De Lit, P., and Delchambre, A., 2000, “Designing Mixed Product Assembly Lines,” *IEEE Journal of Robotics and Automation*, 16(3), pp. 268-280.

CHAPTER 3

CONCURRENT DESIGN OF PRODUCT FAMILIES AND ASSEMBLY SYSTEMS FOR PROFIT MAXIMIZATION

Abstract

To cope with the challenges of market competition and the greater purchasing power of consumers, manufacturers have increased the variety of products. Product families and Reconfigurable Manufacturing Systems (RMS) are used for the cost effective supply of product variety. However, there is a lack of concurrent engineering methods for the joint design of a product family and a RMS since existing concurrent engineering methods were developed for a single product and its associated manufacturing system. The concurrent engineering of a single product and its corresponding manufacturing system is substantially different from the concurrent engineering of a product family and a RMS as the decision space is broader. This chapters introduces a method for concurrently optimizing the design of a product family and a reconfigurable assembly system. The objective of the method is to maximize profits. The problem is formulated as an integer, non-linear program (INLP). A genetic algorithm is used to solve the INLP. The results of the method show that a concurrent

[2]A version of this chapter has been published. Bryan, A., Hu, S. J., and Koren, Y, 2007, "Concurrent Design of Product Families and Assembly Systems," ASME International Manufacturing Science and Engineering Conference, Atlanta, GA, USA. Also submitted to *ASME Journal of Manufacturing Science and Engineering.*

approach to product family and assembly system design leads to solutions that are as good as or better solutions than the sequential approach.

3.1 Introduction

In the early days of manufacturing, product design had to be completed before manufacturing system design could start. To reduce the time and cost required, concurrent engineering methods were introduced to design products and manufacturing systems simultaneously [Boothroyd and Dewhurst, 1989].

More recently, the increase in market competition and the greater purchasing power of consumers have led manufacturers to increase the variety the products they offer. Product families have been introduced for the cost effective supply of variety [Meyer *et al*, 1997]. The common platforms of product families allow manufacturers to acquire economies of scale, while the differentiating components allow manufacturers to meet the needs of diverse consumers [Pine, 1993, Gonzalez *et al*, 2000, Fellini *et al*, 2005, and Jiao *et al*, 2006]. Designing product families involves the determination of the product variants that should be selected for the product family and their corresponding volumes [Li and Azarm, 2002, Green and Krieger, 1985, and Green and Krieger, 1989]. Consumer choice simulators have been commonly used for product variant selection. A comprehensive review of consumer choice simulation techniques is presented in [Green and Krieger, 1989]. The product family design problem is quite challenging to solve since these problems are combinatorial in nature.

The use of product families led to a need for manufacturing systems that can cost effectively produce a variety of products within a single generation and evolve as changes were made to the product family. This led to the introduction of Reconfigurable Manufacturing Systems (RMS). A RMS has been defined by Koren *et al* (1999) as a manufacturing system that is "… designed at the outset for rapid change in structure, as well as in hardware and software components, in order to quickly adjust production capacity and functionality within a part family in response to sudden changes in market or in regulatory requirements".

There is a lack of concurrent engineering methods that can be used for the joint design of product families and reconfigurable manufacturing systems [Michalek *et al*, 2005]. Existing concurrent engineering methods, which were focused on a single product and its corresponding manufacturing system, are substantially different from the concurrent engineering of a product family and a RMS as the decision space for product family and RMS design is broader and different types of decisions need to be made.

This chapter focuses specifically on the concurrent design of product families and reconfigurable assembly systems. The design of assembly systems involves the determination of the system's configuration, the number of workstations required for production, the assignment of tasks to workstations, and the layout of the facility that leads to minimum costs. When simultaneously designing product families and assembly systems, decisions about the number of product variants to include in the product family and the impact of these variants on the design of the assembly system need to be considered [Bryan *et al*, 2007]. This problem was not addressed in the concurrent design of a single product and its associated manufacturing system. Although a few authors have proposed models for the design of product families with consideration of manufacturing costs, their methods do not obtain explicit designs of the manufacturing system [Jiao and Zhang, 2005, Michalek *et al*, 2005, Hernandez *et al*, 2001, Raman *et al*, 1995].

Bryan *et al* (2007) presented a method for the concurrent design of product families and assembly systems. They demonstrated that the combination of product variants selected for a product family affects the design of the assembly system. The model assumed that the demand for product variants can be found by selecting combinations of modules with known demands in such a way that oversupply of modules is minimized and the efficiency of the assembly system is maximized. However, the objective of consumer goods industries is to maximize profits.

The objective of the concurrent product family and assembly system design methodology presented in this chapter is to find the product family and assembly system designs that result in maximum profits. The profit function is computed as the difference between product family revenues and production costs. The product family revenues are found from the selling prices of the product variants which have been selected by consumers. Production costs include the fixed and variable costs associated with the design and operation of the assembly system.

In this chapter, the concurrent product family and assembly system design problem is formulated as an integer, non-linear program (INLP). A genetic algorithm is introduced for the solution of the proposed INLP. Examples are used to demonstrate the implementation of the proposed methodology. It will be demonstrated that a concurrent approach to product family and assembly system design leads to solutions that are as good as or better than those obtained with the sequential approach.

3.2 Product Family Revenue and Cost Models

A product family's profit depends on the revenue that can be obtained from selling the family of products and the production cost for the product family. The first part of this section introduces the terminology and models used to represent the product family. Sections 3.2.2 and 3.2.3 present the models used for computing the product family's revenues and cost of production. Symbols are defined in the Nomenclature section at the end of this chapter.

3.2.1 Product Family Representation

A *product family* is a set of product variants that are constructed from base modules and differentiating modules. *Modules* are the physical components or sub-assemblies through which product features are realized. *Base modules* have one instance that is shared across members of the product family. *Differentiating modules*, have more than one instance that varies among the product variants of the product family. Each *product variant* has exactly one instance of each module. A *null instance* is used to represent the absence of a module from a product variant. Differentiating modules distinguish one product variant from another and are responsible for the variety in a product family.

Modular product architectures are composed of discrete modules and have a one-to-one correspondence between features and modules. An example representation of a modular product architecture is shown in Fig. 3-1. It can be seen that $Module_1$, $Module_2$, and $Module_4$ are base modules, while $Module_3$ and $Module_5$ are differentiating modules. {$Module_1Instance_1$, $Module_2Instance_1$, $Module_3Instance_2$, $Module_4Instance_1$,

$\text{Module}_5\text{Instance}_3\}$ is an example of a product variant that can be formed from the discrete modules shown in Fig. 3-1.

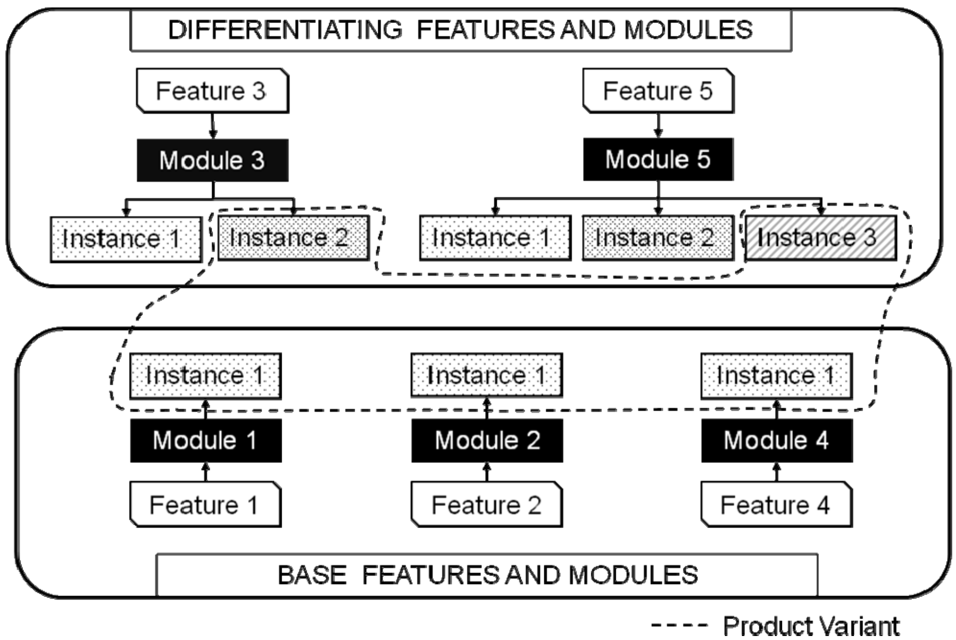


Figure 3-1 Features, modules, and product variant

Denote **J**={1,2,…,*j*,…} as the set of product variants from which the product family is constructed, **K**={1,2,…,*k*,…} as the set of modules used to form product variants, and $\mathbf{L^k}$={1,2,…,ℓ,…} as the set of instances of the k^{th} module. $\mathrm{N} = \prod_{k=1}^{|\mathbf{K}|} |\mathbf{L^k}|$ is the maximum number of product variants that can be formed from a set of modules. Marketing typically sets the limit on the number of product variants that should be included in the product family since not every possible product variant is offered. Given that S is the marketing limit on the number of product variants, the number of possible product families that can be formed from the set of product variants is $\sum_{i=1}^{S} \binom{N}{i} = 2^S - 1$.

One of these product variant combinations is selected as the recommended product family for production. From this computation, it can be seen that the number of possible product families grows rapidly as the number of modules increases.

3.2.2 Product Family Revenue

The revenue of a product family is a function of the set of product variants included in the product family and their corresponding revenues (r_j). Product variant selection is performed in marketing research with the use of consumer choice simulators. A comprehensive review of consumer choice simulators is provided in [Green and Krieger, 1989]. There are three main steps to consumer choice simulation: (1) Determination of consumers' utilities for attributes of products; (2) Prediction of the product variant that will be selected by each consumer; and (3) Computation of the aggregated potential market shares for each product variant.

Experimental design techniques such as conjoint analysis are used to determine consumer utilities for the attributes of a product. *Attributes* are the characteristics of products that influence consumers' selection of products, e.g., the product's size, color, etc. Price is sometimes included as an attribute to be determined. Since the use of conjoint analysis for the determination of consumer utilities is beyond the scope of this research, consumer utilities for product attributes are generated randomly from a uniform distribution.

The next step in consumer choice simulation, the prediction of the product variants selected by consumers, involves estimation of consumer utilities for product variants from the known information of their utilities for attributes. It is assumed that there is a one-to-one mapping between attributes and product functions, and between product functions and instances of modules. Therefore, consumers' utilities for product attributes are considered to be the same as their utilities for module instances. **I**={1,2,…,*i*,…} is considered to be the set of consumers that participate in the marketing analysis and $\omega_{ik\ell}$ represents the utility that consumer i has for instance ℓ of module k. $\omega_{ik\ell}=0$ when a module is not included in a product variant. Consumer *i's* overall utility for a product variant (w_{ij}) is calculated as the difference between the sum of the utilities for the module instances in the product variant and the selling price (revenue) of the product variant as shown in Eq. (3-1).

$$w_{ij} = \left(\sum_{k=1}^{|\mathrm{K}|} \sum_{\ell=1}^{|\mathrm{L}^k|} \omega_{ik\ell} z_{jk\ell} \right) - r_j \quad \forall\, j \in \mathrm{J} \tag{3-1}$$

In Eq. (3-1), $z_{jk\ell}$ is 1 if instance ℓ of module k is in product variant j and 0 otherwise. r_j is the manufacturer's revenue (selling price) for product variant j and is computed as shown in Eq. (3-2).

$$r_j = \sum_{k=1}^{|\mathrm{K}|} \sum_{\ell=1}^{|\mathrm{L}^k|} \rho_{k\ell} z_{jk\ell} \quad \forall\, j \in \mathrm{J} \tag{3-2}$$

where $\rho_{k\ell}$ is the revenue obtained from selling one unit of instance ℓ of module k.

Each consumer is assumed to select either the product variant from the product family that provides the highest overall utility or no product variant at all. The following three equations act as the consumer choice simulator and are used to determine the product variant that is selected by consumer i.

$$w_{ij} - \alpha_i \geq (y_{ij} - 1)\gamma \;\; \forall\, i \in \mathbf{I},\;\; j \in \mathbf{J} \tag{3-3}$$

where y_{ij} is 1 if consumer i selects product variant j and 0 otherwise. α_i is the utility that consumer i associates with a competitive product that is currently available on the market. Its minimum value is zero. γ is a large number. Since the term on the left side of the Eq. (3-3) can be at most w_{ij}, the magnitude of γ is at least as large as the highest consumer utility value.

$$w_{ij} - w_{is} \geq (y_{ij} - 1)\gamma \;\; \forall \;\; i \in \mathrm{I}, s \in \mathrm{J},\;\; j \neq s \tag{3-4}$$

$$\sum_{j=1}^{|\mathrm{J}|} y_{ij} \leq 1 \;\; \forall\, i \in \mathrm{I} \tag{3-5}$$

Eq. (3-3) precludes the selection of a product variant from the product family if there is a product on the market that provides consumer i with a higher overall utility. Eq. (3-4) ensures that if there are multiple product variants that exceed the utility of an existing product, then consumer i will select the product variant with the highest overall utility. Eq. (3-5) limits the maximum number of product variants that consumer i selects to at most 1.

The final step in consumer choice simulation involves the determination of the demand for the product family which depends on the demand for the product variants in the product family. The product variants selected by individual consumers in the previous step are aggregated across consumers to find the demand for each product variant. The market share for product variant j is defined as the ratio of the number of consumers selecting product variant j to the total number of consumers surveyed, $\frac{\sum_{i=1}^{|\mathbf{I}|} y_{ij}}{|\mathbf{I}|}$. Assuming that there are ψ potential consumers in the market, the total demand for product variant j (v_j) is determined from the following equation.

$$v_j = \frac{\psi}{|\mathbf{I}|} \sum_{i=1}^{|\mathbf{I}|} y_{ij} \tag{3-6}$$

Using the product variant revenues (r_j) and the total demand for product variants (v_j), the overall revenue (Θ_1) of the product family is computed as follows:

$$\Theta_1 = \sum_{j=1}^{|\mathbf{J}|} r_j v_j \tag{3-7}$$

3.2.3 Production Cost for the Product Family

The cost of production for the product family is a function of the fixed and variable costs of the assembly system. The fixed costs are the investment costs associated with the purchase and installation of new equipment. The computation of variable costs can be complex as variable costs consist of all the costs associated with the running of a factory such as labor costs, utility costs, repair costs, material handling system costs, etc. Only the labor costs associated with the running of workstations are considered in this research.

A *center* is defined as the set of physical tools and fixtures required to complete a set of tasks. The number of centers required is dependent on the configuration of the assembly system. The assembly system is assumed to consist of serial groups of parallel centers where each group of parallel centers is a *workstation*. The total number of centers is the sum of the number of centers required at each workstation. Therefore, fixed and variable costs of the assembly system are a function of the number of centers required for assembly.

Since multiple product variants are assumed to be assembled simultaneously on a serial line, the assembly system is referred to as a *mixed model assembly line*. The total number of workstations and centers required at each workstation is determined by solving the mixed model assembly line balancing problem (MALBP) [Becker and Scholl, 2006]. The product variants and their corresponding demand values obtained from consumer choice modeling are used as inputs to the MALBP. The MALBP is solved in two steps. The product family precedence diagram is obtained in the first step [Becker and Scholl, 2006, Thomopoulos, 1970, Thomopoulos, 1967]. Tasks are assigned to workstations in the next step.

Precedence diagrams are directed graphs that represent the order of task completion. Each node in traditional precedence diagrams represents the joining of two modules. The traditional representation is modified by the addition of a null task to the beginning of the precedence diagram. The null task has a task time of zero and represents the introduction of the first module to the assembly line. With this modification, each node represents the joining of one module to the previously completed subassembly. Therefore, consistency between the index of an assembly task and the index of the module being assembled is achieved.

All product variants in a product family are assumed to have all tasks. The difference in the assembly times of the instances of differentiating modules results in a unique precedence diagram for each product variant e.g. Fig 3-2 (a-b). When an assembly task is not required for a particular product variant, the task is assigned a time of zero, e.g. task 8 in Fig. 3-2a. The product family precedence diagram is developed by combining the precedence diagrams of the individual product variants and computing the product family assembly time for the tasks in the combined representation.

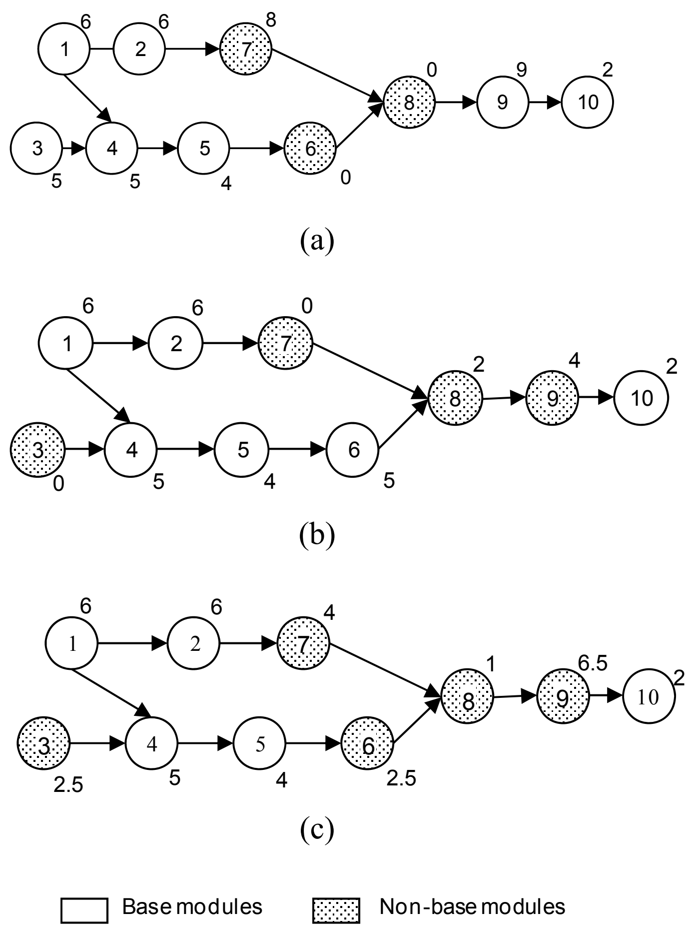


Figure 3-2 (a) Precedence diagram for product variant 1; (b) Precedence diagram for product variant 2; (c) Product family precedence diagram

The product family assembly time for a given module k, t_k^{PF}, is the weighted average of the task times required for each product variant and is computed as follows

$$t_k^{PF} = \sum_{j=1}^{|\mathrm{J}|} \sum_{\ell=1}^{|\mathrm{L}^k|} \tau_{k\ell} z_{jk\ell} p_j \tag{3-8}$$

where $\tau_{k\ell}$ is the time required for the assembly of instance ℓ of module k, and p_j is the proportion of production volume that is attributed to product variant j. p_j is calculated by dividing the required volume for product variant j by the total volume of product variants, Eq. (3-9). As an illustration, assume that the product family consists of the two product variants shown in Fig. 3-2(a-b) with a production ratio of 50% for each product variant. The precedence diagrams of these product variants are combined to form a product family precedence diagram as shown in Fig. 3-2c.

$$p_j = \frac{v_j}{\sum_{j=1}^{|\mathrm{J}|} v_j} \tag{3-9}$$

Once the product family precedence diagram is determined, the next step in solving the MALBP involves the assignment of tasks to workstations. The conditions for task assignment are: (1) each task must get assigned; (2) the weighted average assembly time of tasks assigned to a workstation must not exceed the capacity at the workstation; and (3) precedence constraints must not be violated.

The traditional task indivisibility constraint ensures that every task is assigned to a workstation.

$$\sum_{m=1}^{|\mathrm{M}|} x_{km} = 1 \quad \forall \ k \in \mathrm{K} \tag{3-10}$$

The average load at workstation m, $\sum_{k=1}^{|\mathbf{K}|} t_k^{PF} x_{km}$, is a function of the tasks assigned to the workstation and the product family assembly time of the tasks. The available capacity at workstation m is $C \times b_m$ where C is computed as in Eq. (3-11). Eq. (3-12) gives the workload feasibility requirement at a given workstation.

$$C = \frac{\lambda}{\sum_{j=1}^{|\mathrm{J}|} v_j} \tag{3-11}$$

$$\sum_{k=1}^{|\mathrm{K}|} t_k^{PF} x_{km} \leq C b_m \quad \forall \ m \in \mathrm{M} \tag{3-12}$$

The precedence constraint for task assignment is given in Eq. (3-13).

$$x_{km} \leq \sum_{n=1}^{m} x_{qn} \quad \forall \ n, m \in \mathrm{M}, \ k \in \mathrm{K}, \ q \in \mathrm{P}(k) \tag{3-13}$$

where P(k) is the set of predecessors of task k.

The number of parallel centers (b_m) at each workstation is obtained by solving the above line balancing equations. Assuming that exactly one worker is required at each workstation, that the life of the product family is known in advance, and that all workstations have the same investment cost, the production cost of the system (Θ_2) is computed as follows:

$$\Theta_2 = \sum_{m=1}^{|M|} (\phi + \upsilon\lambda) b_m \tag{3-14}$$

where ϕ is the fixed cost per center, υ is the wage rate per worker per unit time and λ is the life of the assembly system.

3.3 Optimization Formulation

The decision variables for the concurrent product family and assembly system design problem are b_m, x_{km}, y_{ij}, and $z_{jk\ell}$. b_m is the number of parallel centers at workstation m. A value of 0 for b_m means that workstation m is not utilized. The stations are utilized in increasing values of m. x_{km} indicates the workstation that module k is assigned to for assembly. x_{km} is 1 if module k is assembled at workstation m and 0 otherwise. y_{ij} indicates the product variant selected by a consumer i. y_{ij} is 1 if consumer i selects product variant j and 0 otherwise. $z_{jk\ell}$ indicates that instance ℓ of module k is selected for product variant j.

The complete model for the concurrent product family and assembly system design problem is as follows:

$$\text{maximize } \Theta = \Theta_1 - \Theta_2 = \sum_{j=1}^{|\mathrm{J}|} r_j v_j - \sum_{m=1}^{|\mathrm{M}|} (\upsilon\lambda + \phi) b_m \tag{3-15}$$

subject to:

$$\sum_{\ell=1}^{|\mathrm{L^k}|} z_{jk\ell} = 1 \quad \forall j \in \mathrm{J},\ k \in \mathrm{K} \tag{3-16}$$

$$w_{ij} - \alpha_i \geq (y_{ij} - 1)\gamma \quad \forall\, i \in \mathrm{I},\ j \in \mathrm{J} \tag{3-17}$$

$$w_{ij} - w_{is} \geq (y_{ij} - 1)\gamma \quad \forall \ i \in \mathrm{I}, s \in \mathrm{J}, \ j \neq s \tag{3-18}$$

$$\sum_{j=1}^{|\mathrm{J}|} y_{ij} \leq 1 \ \ \forall i \in \mathrm{I} \tag{3-19}$$

$$\sum_{m=1}^{|\mathrm{M}|} x_{km} = 1 \quad \forall \ k \in \mathrm{K} \tag{3-20}$$

$$\sum_{k=1}^{|\mathrm{K}|} \sum_{\ell=1}^{|\mathrm{L}^k|} \sum_{j=1}^{|\mathrm{J}|} \tau_{k\ell} z_{jk\ell} x_{km} \leq \lambda b_m \quad \forall \ m \in \mathrm{M} \tag{3-21}$$

$$x_{km} \leq \sum_{n=1}^{m} x_{qn} \quad \forall n, m \in \mathrm{M}, q \in \mathrm{P}(k) \tag{3-22}$$

$$where \, b_m \geq 0, \ x_{km} \in \{0,1\}, \ y_{ij} \in \{0,1\}, \ z_{jk\ell} \in \{0,1\} \tag{3-23}$$

The objective of the product family and assembly system design problem, Eq. (3-15) is to maximize profit over the period that the manufacturer desires to have the product family in production. Profit (Θ) is computed as the difference between product family revenue (Θ_1) and production cost (Θ_2). The formulations for Θ_1 and Θ_2 are given in Eq. (3-7) and Eq. (3-14) respectively.

The constraints in Eq. (3-16 to 3-19) are associated with product variant selection while the remaining constraints, Eq. (3-20 to 3-22), are associated with MPALB. The constraint in Eq. (3-16) ensures that exactly one instance of each module is assigned to each product variant. The constraints in Eq. (3-17 to 3-19) are the consumer choice simulation equations from Eq. (3-3 to 3-5) and Eq. (3-20 to 3-22) are the MPALB constraints from Eq. (3-10, 3-12 to 3-13). These equations were repeated in this section for completeness. The feasibility constraints on the decision variables are given in Eq. (3-23).

3.4 Solution Approach

The mathematical model presented in the previous section for finding the profit maximizing product family and assembly system is an integer, non-linear program (INLP). INLP's are combinatorial problems which are difficult to solve by exact optimization methods. Genetic algorithms (GAs), which were first introduced by Holland (1975), are global search algorithms that have been successfully used for the solutions of these types of problems. GA's can easily handle integer, non-linear, non-convex functions. Although the optimality of the solutions cannot be proven, GAs have been shown to provide good results in solving both product family design and assembly line balancing problems [Balakrishnan and Jacob, 1996, Leu *et al*, 1994].

The four main stages of formulating a GA are: (1) Encoding and decoding of the solution; (2) Formation of the initial population; (3) Selection of members of the population for reproduction; and (4) Genetic manipulation. The techniques for encoding and solving the GA used in this paper are adaptations of previous methods presented by Balakrishnan and Jacob (1996) and Leu *et al* (1994).

3.4.1 Chromosome Representation, Encoding, and Decoding

The genes in this problem are real integer variables. A multi-sectioned chromosome string is used to represent a solution to this problem. An example of a chromosome string for a product family with two product variants is shown in Fig. 3-3.

The total number of sections in a chromosome is $|\mathbf{J}|+1$. The first $|\mathbf{J}|$ sections represent a product family where each j^{th} subsection represents a product variant in the product family. The $\left(|\mathbf{J}|+1\right)^{\text{st}}$ section represents the task sequence used for the MALBP.

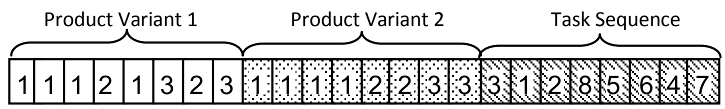


Figure 3-3 Chromosome representation

Each gene in a product variant subsection of a chromosome represents a module with the position of the gene indicating the index of the module. For example, the first gene position represents the first module, the second gene position, the second module, etc. Therefore, the total number of genes in a product variant subsection of a chromosome is equal to the cardinality of set **K**. The value in the gene is the instance of the module that is selected for the particular product variant. In Fig. 3-3, it can be seen that the 3$^{\text{rd}}$ instance is selected for the 6$^{\text{th}}$ module in the 1$^{\text{st}}$ product variant. This chromosome representation of genes is consistent with the constraint in Eq. (3-16).

The last subsection of a chromosome represents a task sequence and is inspired by the representation used by Leu (1994). The position of each gene represents the order of a task in the sequence, while the value in the gene represents the module that is in the given position. Fig. 3-4 shows the decoding of assembly tasks into workstations. The tasks are selected in the order represented by this sequence to be placed into workstations. Then, the number of parallel centers required to assemble the assigned tasks is computed. This encoding/decoding of tasks fulfills the requirements of the constraints in Eq. (3-20 to 3-21).

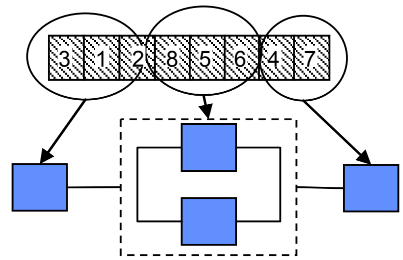


Figure 3-4 Decoding of the task sequence into workstations

Another advantage of using the chromosome representation in Fig. 3-3 is that it allows for the determination of the optimal number of product variants that should be included in the product family. This is accomplished in two ways. Firstly, all the subsections in the chromosome may not be unique. Therefore, although the chromosome may contain sufficient subsections to represent the maximum number of product variants in the product family, by the replication of subsections, different chromosomes may represent differing number of product variants. Secondly, some of the subsections may have zero consumer demand. Therefore, the optimal number of product variants for the product family can be determined.

3.4.2 Initial Population

The initial population is randomly generated. Since, the number of potential instances for a given module can differ from that of other modules in a product variant, the value for the instance of each gene is individually generated. The line balancing subsection of the chromosome is generated by obtaining a random permutation of the modules as shown in Fig. 3-3. This randomized order represents one potential assembly sequence of modules.

3.4.3 Selection

Chromosomes are selected for genetic reproduction (crossover) based on their fitness values. There are several selection methods. This research uses an elitist selection approach in which the top 50% of chromosomes are selected for crossover. The fitness function consists of the objective function, Eq. (3-15), and a penalty for precedence constraint violations as follows:

$$fitness = \sum_{j=1}^{|J|} r_j v_j - \sum_{m=1}^{|M|} (\upsilon\lambda + \phi) b_m - \kappa b_m \tag{3-24}$$

where κ is a number that is selected to be large enough to provide a suitable penalty for infeasible line balancing solutions. It is determined by trial and error. A value of κ between two and ten times the total cost of a center has provided suitable penalties for the problems tried in this research. This penalty significantly reduces the fitness value of a chromosome, thereby ensuring that solutions violating the precedence constraint in Eq. (3-22) have a very low possibility of survival.

3.4.4 Genetic Manipulation

Once the parents are selected, genetic manipulation is performed. The genetic operators used are crossover, mutation and inversion. Two rates are associated with each of the operators. The first rate determines the number of generations that the operator is used and the second rate determines the number of members in the population of each generation that undergoes genetic manipulation.

Two point crossover within the subsections is used for both the product variant and line balancing subsections. In the first step of crossover, the parent chromosomes are divided into their individual subsections. For each subsection, the positions of two crossover sites are randomly generated. The parent genes between the two crossover sites are then exchanged to form two new subsections as shown in Fig. 3-5. For the line balancing subsection, a check is made to determine whether the newly inserted genes contain tasks that are redundant with the genes that are not exchanged. If redundant tasks are found, the redundant tasks are deleted from the original child subsection. The line balancing subsection is then repaired by randomly assigning a task to the gene position ensuring that there are no redundancies. The children subsections are then recombined to form the new chromosome.

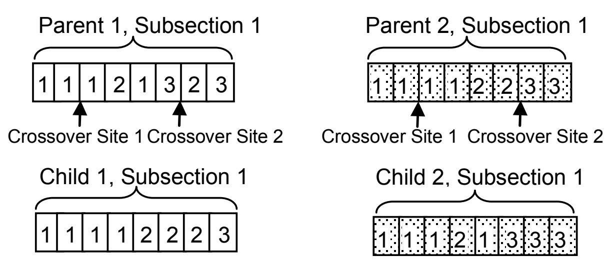


Figure 3-5 Crossover operator

As for crossover, mutation also occurs within subsections. The number and location of the mutation sites of each subsection are randomly generated. The values in the selected sites are set to zero. A new value for the gene subsection is then obtained by selecting a feasible value randomly, Fig. 3-6. For product variant subsections, this involves randomly generating a feasible value for the instance of the module. For the line balancing subsection, this involves generating a new task into the mutation site ensuring that there are no task redundancies.

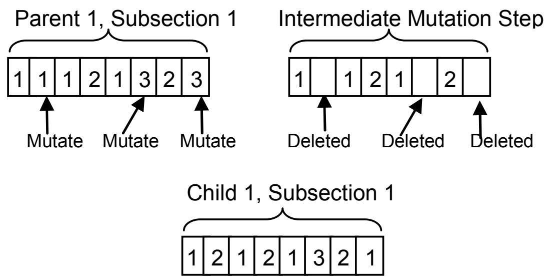


Figure 3-6 Mutation operator

Inversion is only performed for product variant subsections of the chromosome. The inversion operator involves reversal of the order in which product variant subsections appear in the chromosome as shown in Fig. 3-7. This operator changes the appearance of the chromosome but not the fitness value. Use of the inversion operator increases the diversity in the population.

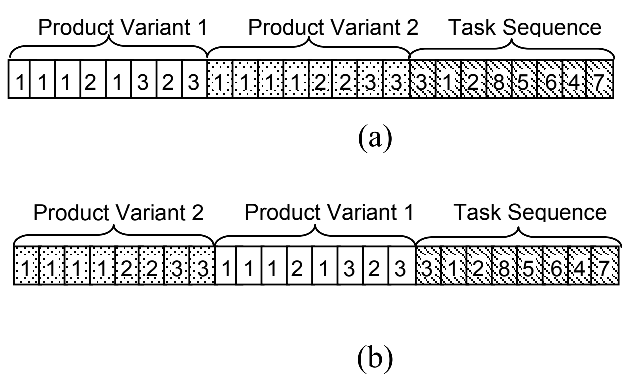


Figure 3-7 Inversion operator (a) Before inversion (b) After inversion

3.5 Examples

Three examples are presented in this section. The first example is a detailed explanation of the application of the concurrent product family and assembly system design methodology to the design of the product family and assembly system of an office chair. Two other examples demonstrate the implementation of the approach to larger problems identified from the literature. In addition to demonstrating the application of the proposed methodology, the examples demonstrate the advantage of using the concurrent product family and assembly system design methodology over the existing sequential approach. This section concludes with a discussion of the computational efficiency of the methodology.

3.5.1 Example 1

The objective of this example is to design a family of office chairs and their accompanying mixed model assembly system. The office chair consists of nine modules, six base modules and three differentiating modules as shown in Fig. 3-8.

The number of instances for each module and their corresponding assembly times are provided in Table 3-1. In this example, instance 1 of the differentiating modules are null modules, which means that these modules are not included in the product variants. The precedence diagram for assembly is shown in Fig. 3-9. The assembly times for the product family precedence diagram are computed using Eq. (3-8). The number of product variants (N) is 12 and the number of potential product families (S) is 4095. One of these 4095 product variant combinations is selected as the most profitable product family. The remaining parameters used for this example are provided in Table 3-2. The customer utility values are in Appendix 3A and the revenues are in Appendix 3B.

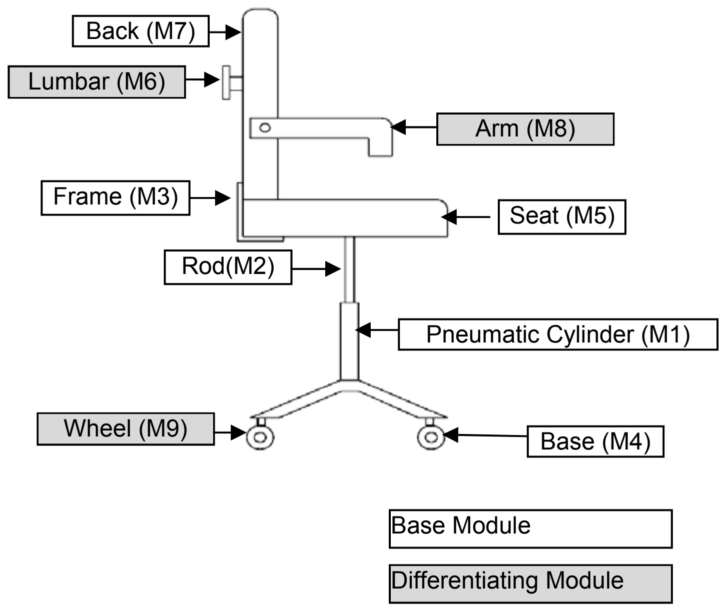


Figure 3-8 Office chair showing modules

Table 3-1 Office chair modules and assembly times

Module	**M1**	**M2**	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**	**M9**
# of Instances	1	1	1	1	1	2	1	2	3
Assembly Time Instance 1 (min.)	0	8	16	6	14	0	8	0	0
Assembly Time Instance 2 (min.)	-	-	-	-	-	12	-	10	8
Assembly Time Instance 3 (min.)	-	-	-	-	-	-	-	-	16

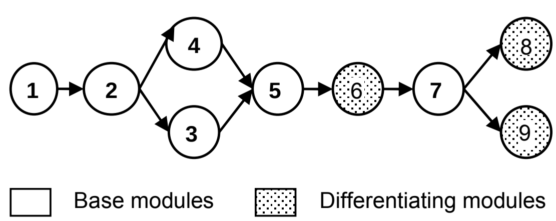


Figure 3-9 Office chair precedence diagram

Table 3-2 Office chair parameters

Parameter	Value
Product Life	2.34×10^5 min
# of Consumers Interviewed	25
Market Size	25, 000
Fixed Workstation Cost	$50K
Labor Cost	$20/hr
Consumer worth / attribute: U($0,$40)	
Selling price / attribute: U($1,$35)	

This example is solved by both the genetic algorithm presented in Section 3.4 and the sequential approach discussed in Section 3.1. The results for maximum profit, product family revenues and assembly system costs from both methodologies are compared. The product variants selected for the product family and assembly line design solution are reported for both methodologies.

For the genetic algorithm, a population size of 64 chromosomes was maintained in each generation. Crossover was performed for all generations. Mutation was performed for 50% of the generations and inversion for 25%. 10% and 25% of the members of each population underwent mutation and inversion respectively. The termination criterion was set to 100 generations.

The profit, market share, and costs for the concurrent and sequential approaches are shown in Table 3-3. These results show that the sequential approach found the product family with higher revenues than the concurrent approach. However, the selected product family had higher costs. Therefore, a greater profit is obtained with the concurrent approach.

Table 3-3 Results for the concurrent and the sequential approaches

	Concurrent Approach	Sequential Approach
Profit	\$2.32M	\$2.29M
Revenue	\$3.34M	\$3.44M
Cost	\$ 1.02M	\$1.15M
No. of Product Variants	2	2
Market Share	84%	84%
No. of Centers	8	9

For both the sequential and concurrent approaches, the selected product family has 2 product variants and captures 84% market share, Table 3-4 a-b. Since both approaches captured the same market share, the difference in production costs is due to the differences in the product family assembly times of modules in the product variants. The differences in revenues were also due to the differences in the revenues of the product variants selected for the product family.

Table 3-4 Product family assembly times for office chair (a) Concurrent approach (b) Sequential approach

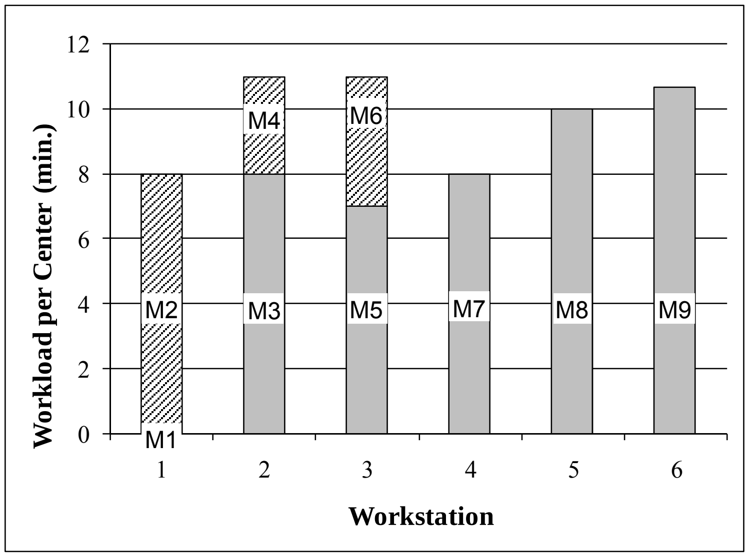
Product Variant	Modules			Market Share	Product Variant Revenue
	M6	M8	M9		
1	1	2	3	0.28	\$1.09M
2	2	2	2	0.56	\$2.25M

(a)

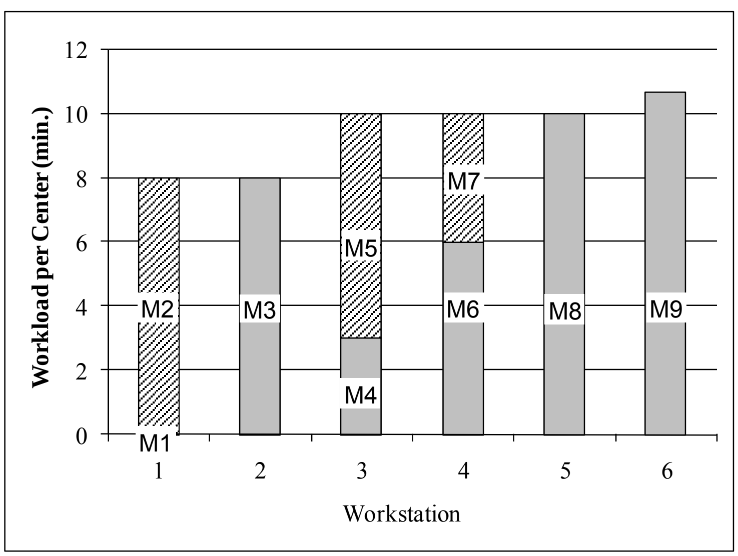
Product Variant	Modules			Market Share	Product Variant Revenue
	M6	M8	M9		
1	2	2	3	0.28	\$1.18M
2	2	2	2	0.56	\$2.25M

(b)

The workstation loads for both the concurrent and sequential approaches and their corresponding assembly system configurations are shown in Fig. 3-10 and Fig. 3-11 respectively. The task assignments for the concurrent and sequential approach are also shown in Fig. 3-10.

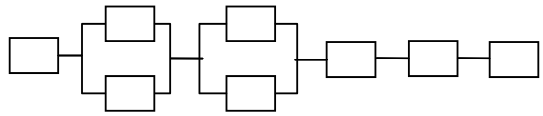


(a)

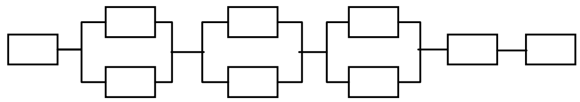


(b)

Figure 3-10 Workstation loads



(a)



(b)

Figure 3-11 Assembly line configurations

3.5.2 Larger Examples

The concurrent product family and assembly system design methodology was applied to two larger examples using modifications of existing precedence diagrams commonly found in the assembly line balancing literature. The modules with instances, their corresponding assembly times and other details of the necessary to solve the problem were randomly generated. The inputs to these problems are provided in Appendix 3C. The first example has 11 modules and 18 product variants [Becker and Scholl, 2006]. The second example, which has 30 modules and 32 product variants, is an example from an automotive assembly line [Sawyer, 1970]. The genetic algorithm from Section 3-4 was used to find the product family and assembly system designs for these two examples. The details of the genetic algorithm are shown in Table 3-5.

These problems were also solved using the sequential approach discussed previously. A comparison of the results between the concurrent approach and the sequential approach are given in Table 3-6. For the 11 module problem, the profit obtained from the concurrent approach is higher than that obtained with the sequential approach. The maximum profit solution for the for the 30 task problem using the sequential approach could not be found in a reasonable amount of time due to a lack of sufficient computer memory.

Table 3-5 Genetic algorithm parameters for example2 and example 3

	Example 2 11 Modules 18 Product Variants	Example 3 30 Modules 32 Product Variants
Population Size	200	500
Mutation Rate	0.5	0.5
Inversion Rate	0.25	0.1
Number of Generations	1000	1000

Table 3-6 Results for example 2 and example 3

	Example 2 11 Modules 18 Product Variants		Example 3 30 Modules 32 Product Variants	
	Concurrent Approach	**Sequential Approach**	**Concurrent Approach**	**Sequential Approach**
Profit	$28.7M	$27.7M	$56.1M	Solution not found
Revenue	$35.9M	$36.1M	$93.3M	
Cost	$ 7.2M	$ 8.4M	$37.2M	
No. of Product Variants	7	5	3	
Market Share	62%	62%	68%	
No. of Centers	6	7	31	

3.5.3 Discussion

From the results of the three examples, it is observed that the concurrent product family and assembly system design methodology leads to solutions as good as or better than could be obtained with the concurrent approach. The results indicate that the product family with the maximum profit does not necessarily have the highest revenue. The sequential approach inherently assumes that a solution with the maximum revenue will have the maximum profit. Therefore, it was not able to find solutions as good as the concurrent approach which is not limited by this assumption.

The examples were solved in Matlab Version 7.0 on a PC with a Intel Pentium, 2 GHz processor. The GA was able to find the reported solutions to the above three examples in 4.6 sec, 55 sec and 4620 sec respectively. The sequential approach obtained solutions in 4.6 sec and 283 sec respectively for the 1st and 2nd examples. The number of combinations needed to be checked for 30 module problem was so great that a solution was not attainable in a reasonable amount of time on the computer used. Therefore, in addition to finding better solutions, the concurrent approach presented in this paper is more computationally efficient than the sequential approach.

3.6 Conclusions

In this chapter a method for the concurrent design of a product family and assembly system is presented. It allows for the selection of product variants for the product family. In addition, the number of required workstations and the assignment of tasks to workstations are determined. Examples were used to show that the product family that leads to maximum revenues does not necessarily lead to maximum profits. Therefore, the concurrent product family and assembly system design method finds solutions as good as or better than the sequential approach. In addition to finding good solutions, the genetic algorithm used for the concurrent product family and assembly system problem is more computationally efficient than the sequential approach.

Acknowledgements

The author would like to acknowledge the support of the Engineering Research Center for Reconfigurable Manufacturing Systems at the University of Michigan (NSF Grant No. EEC95-92125). The author also acknowledges co-authorship with Prof. S. Jack Hu and Prof. Yoram Koren. In addition, we would like to thank Dr. Jeff Abell from the General Motors Corporation for his numerous contributions.

Nomenclature

Sets

I	$=\{1, 2, \ldots, i, \ldots\}=$	set of consumers used for consumer choice simulation
J	$=\{1, 2, \ldots, j, \ldots\}=$	set of product variants
K	$=\{1, 2, \ldots, k, \ldots\}=$	set of product modules
$\mathbf{L^k}$	$=\{1, 2, \ldots, \ell, \ldots\}=$	set of instances of module k
M	$=\{1, 2, \ldots, m, \ldots\}=$	set of workstations
P(k)	=	set of predecessors to module k

Variables

b_m	=	number of parallel centers at workstation m
C	=	cycle time
p_j	=	proportion of production that is product variant j
r_j	=	revenue from one unit of product variant j ($)
t_k^{PF}	=	weighted average assembly time for module k
v_j	=	total production volume of product variant j
w_{ij}	=	utility that consumer i derives from product variant j ($)
x_{km}	=	assignment variable for module k to workstation m
y_{ij}	=	assignment variable of product variant j to consumer i
$z_{jk\ell}$	=	assignment variable for instance ℓ of module k to product

Parameters

Θ = Profit ($)

Θ_1 = product Family Revenue ($)

Θ_2 = product Family Production Cost ($)

α_i = utility that consumer i associates with a currently available product ($)

ϕ = fixed cost per workstation ($)

γ = a large number that is used in the selection of product variants

κ = a large number that is used to compute the fitness function

λ = expected life of the product family

Π = Profit ($)

$\rho_{k\ell}$ = revenue from selling one unit of instance ℓ of module k ($)

$\tau_{k\ell}$ = time required to assemble instance ℓ of module k

υ = variable cost per workstation per unit time

ω_{ik} = utility of instance ℓ of module k to the i^{th} consumer ($)

ψ = total number of potential consumers in the market

APPENDIX 3A

Table 3A-1 Data of customer utility values for example 1

		M1	M2	M3	M4	M5	M6	M7	M8	M9
Customer 1	**Option 1**	16	23	5	1	9	0	15	0	0
	Option 2	-	-	-	-	-	28	-	19	9
	Option 3	-	-	-	-	-	-	-	-	31
Customer 2	**Option 1**	20	39	37	16	7	0	9	0	0
	Option 2	-	-	-	-	-	23	-	7	9
	Option 3	-	-	-	-	-	-	-	-	27
Customer 3	**Option 1**	27	21	33	12	17	0	25	0	0
	Option 2	-	-	-	-	-	32	-	14	10
	Option 3	-	-	-	-	-	-	-	-	5
Customer 4	**Option 1**	0	3	35	13	34	0	7	0	0
	Option 2	-	-	-	-	-	38	-	9	11
	Option 3	-	-	-	-	-	-	-	-	23
Customer 5	**Option 1**	16	2	35	25	37	0	34	0	0
	Option 2	-	-	-	-	-	19	-	32	38
	Option 3	-	-	-	-	-	-	-	-	37
Customer 6	**Option 1**	7	30	7	11	17	0	27	0	0
	Option 2	-	-	-	-	-	8	-	15	31
	Option 3	-	-	-	-	-	-	-	-	8
Customer 7	**Option 1**	38	0	20	27	34	0	32	0	0
	Option 2	-	-	-	-	-	1	-	39	28
	Option 3	-	-	-	-	-	-	-	-	4
Customer 8	**Option 1**	8	33	4	17	20	0	19	0	0
	Option 2	-	-	-	-	-	37	-	11	26
	Option 3	-	-	-	-	-	-	-	-	8
Customer 9	**Option 1**	14	33	25	27	26	0	35	0	0
	Option 2	-	-	-	-	-	28	-	25	35
	Option 3	-	-	-	-	-	-	-	-	13
Customer 10	**Option 1**	22	28	29	38	22	0	9	0	0
	Option 2	-	-	-	-	-	13	-	19	4
	Option 3	-	-	-	-	-	-	-	-	28
Customer 11	**Option 1**	9	19	32	36	7	0	26	0	0
	Option 2	-	-	-	-	-	36	-	26	38
	Option 3	-	-	-	-	-	-	-	-	37
Customer 12	**Option 1**	14	18	11	27	10	0	22	0	0
	Option 2	-	-	-	-	-	22	-	1	37
	Option 3	-	-	-	-	-	-	-	-	38
Customer 13	**Option 1**	12	15	8	38	4	0	22	0	0
	Option 2	-	-	-	-	-	28	-	26	15
	Option 3	-	-	-	-	-	-	-	-	40
Customer 14	**Option 1**	17	5	35	27	11	0	12	0	0
	Option 2	-	-	-	-	-	35	-	21	3
	Option 3	-	-	-	-	-	-	-	-	33
Customer 15	**Option 1**	4	14	32	30	21	0	26	0	0
	Option 2	-	-	-	-	-	10	-	30	8
	Option 3	-	-	-	-	-	-	-	-	20
Customer 16	**Option 1**	31	12	37	15	28	0	16	0	0
	Option 2	-	-	-	-	-	1	-	26	13
	Option 3	-	-	-	-	-	-	-	-	28
Customer 17	**Option 1**	28	6	13	25	21	0	0	0	0
	Option 2	-	-	-	-	-	16	-	32	24
	Option 3	-	-	-	-	-	-	-	-	3

Table 3A-1 Continued

		M1	M2	M3	M4	M5	M6	M7	M8	M9
Customer 18	**Option 1**	36	28	21	14	25	0	28	0	0
	Option 2	-	-	-	-	-	2	-	17	34
	Option 3	-	-	-	-	-	-	-	-	27
Customer 19	**Option 1**	15	35	34	32	18	0	25	0	0
	Option 2	-	-	-	-	-	27	-	7	14
	Option 3	-	-	-	-	-	-	-	-	10
Customer 20	**Option 1**	10	9	26	27	29	0	25	0	0
	Option 2	-	-	-	-	-	15	-	37	21
	Option 3	-	-	-	-	-	-	-	-	10
Customer 21	**Option 1**	15	12	9	15	21	0	32	0	0
	Option 2	-	-	-	-	-	23	-	7	24
	Option 3	-	-	-	-	-	-	-	-	17
Customer 22	**Option 1**	15	12	17	0	30	0	39	0	0
	Option 2	-	-	-	-	-	23	-	38	11
	Option 3	-	-	-	-	-	-	-	-	15
Customer 23	**Option 1**	20	18	40	6	26	0	18	0	0
	Option 2	-	-	-	-	-	39	-	28	31
	Option 3	-	-	-	-	-	-	-	-	16
Customer 24	**Option 1**	18	35	25	15	33	0	26	0	0
	Option 2	-	-	-	-	-	17	-	5	3
	Option 3	-	-	-	-	-	-	-	-	36
Customer 25	**Option 1**	1	34	30	27	15	0	3	0	0
	Option 2	-	-	-	-	-	2	-	21	28
	Option 3	-	-	-	-	-	-	-	-	12

APPENDIX 3B

Table 3B-1 The revenues for module-instances for example 1

	M1	M2	M3	M4	M5	M6	M7	M8	M9
Instance 1	48	48	48	48	48	0	48	0	0
Instance 2	0	0	0	0	0	41	0	39	32
Instance 3	0	0	0	0	0	0	0	0	30

APPENDIX 3C

Table 3C-1 Parameters for examples 2 and 3

Parameter	Value
Product Life	6.05×10^5 min
# of Consumers Interviewed	25
Market Size	100,000
Fixed Workstation Cost	$1M
Labor Cost	$20/hr
Consumer worth / attribute: U($0,$100)	
Selling price / attribute: U($5,$105)	

Table 3C-2 Example 2 modules and assembly times

Module	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
# of Instances	1	1	1	1	1	1	1	1	2	3	3
Assembly Time Instance 1 (min.)	0	6	6	5	5	4	5	4	0	0	0
Assembly Time Instance 2 (min.)	-	-	-	-	-	-	-	-	2	9	2
Assembly Time Instance 3 (min.)	-	-	-	-	-	-	-	-	-	3	5

Table 3C-3 Example 3 modules and assembly times

Module	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
# of Instances	1	1	1	1	1	1	1	1	1	1	1
Assembly Time Instance 1 (min.)	8	7	19	10	2	6	14	10	1	4	14
Assembly Time Instance 2 (min.)	-	-	-	-	-	-	-	-	-	-	-

Module	M12	M13	M14	M15	M16	M17	M18	M19	M20	M21	M22
# of Instances	1	1	1	1	1	1	1	1	2	2	1
Assembly Time Instance 1 (min.)	15	5	12	9	10	2	10	18	16	21	-
Assembly Time Instance 2 (min.)	-	-	-	-	-	-	-	-	7	0	12

Module	M23	M24	M25	M26	M27	M28	M29	M30			
# of Instances	1	2	1	2	1	2	1	1			
Assembly Time Instance 1 (min.)	-	7	17	9	25	7	-	-			
Assembly Time Instance 2 (min.)	20	4	-	0	-	0	18	1			

Table 3C-4 Precedence table for example 2

0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0	0
1	1	0	1	1	0	0	0	0	0	0
1	1	0	1	1	1	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	0	0	0
1	1	1	1	1	1	1	1	1	0	0
1	1	1	1	1	1	1	1	1	1	0

Table 3C-5 Precedence table for example 3

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	1	0	1	1	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	1	0	1	1	1	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	1	0	0	0	1	0	0	0	1	1	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	0	0	0
1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	0	0	0
1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	0	1	0

References

Balakrishnan, P. V., and Jacob, V. S., 1996, "Genetic Algorithms for Product Design," *Management Science*, 42(8)pp. 1105-1117.

Becker, C., and Scholl, A., 2006, "A Survey on Problems and Methods in Generalized Assembly Line Balancing," *European Journal of Operational Research*, 168(3), pp. 694-715.

Boothroyd, G., and Dewhurst, P., 1989, *Product Design for Assembly*, Wakefield, RI.

Bryan, A., Ko, J., Hu, S. J., and Koren, Y., 2007, "Co-Evolution of Product Families and Assembly Systems," *CIRP Annals*, 56(1), pp. 21-44.

Bryan, A., Hu, S. J., Koren, Y., 2007, "Concurrent Product Portfolio Planning and Mixed Product Assembly Line Balancing," *Chinese Journal of Mechanical Engineering*, 20(1), pp. 96-99.

Fellini, R., Kokkolaras, M., Papalambros, P., and Perez-Duarte, A., 2005, "Platform Selection Under Performance Bounds in Optimal Design of Product Families," *ASME Journal of Mechanical Design*, 37(4), pp. 524-535.

Gonzalez-Zugasti, J., Otto, K., and Baker, J., 2000, "A Method for Architecting Product Platforms," *Research in Engineering Design*, 12(2), pp.61-72.

Green, P., and Krieger, A., 1985, "Models and Heuristics for Product Line Selection," *Marketing Science*, 4(1), pp.1-19.

Green, P., and Krieger, A., 1989, "Recent Contributions to Optimal Product Positioning and Buyer Segmentation," *European Journal of Operations Research*, 41(2), pp.127-141.

Hernandez, G., Allen, J., Simpson, T., Bascaran, E., Avila, L., and Salina, F., 2001, "Robust Design of Families of Products with Production Modeling and Evaluation," *ASME Journal of Mechanical Design*, 123(2), pp.183-190.

Holland, J. H., 1975, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, Michigan, USA.

Jiao, J., Simpson, T. W., and Siddique, Z., 2006, "Product Family Design and Product-Based Platform Development: A State-of-the-Art Review," *Journal of Intelligent Manufacturing*, Special Issue, pp. 1-36.

Jiao, J., and Zhang, Y, 2005, "Product Portfolio Planning with Customer-Engineering Interactions," *IIE Transactions*, 37(9), pp. 801-814.

Koren, Y., Heisel, U., Jovan, F., Moriwaki, T., Pritschow, G., Ulsoy, G., and Van Brussel, H., 1999, "Reconfigurable Manufacturing Systems," *CIRP Annals*, 48(2), pp. 527-540.

Leu, Y., Matheson, L., and Rees, L., 1994, "Assembly Line Balancing Using Genetic Algorithms with Heuristic Generation Initial Population and Multiple Evaluation Criteria," *Decision Sciences*, 25(4), pp. 581-606.

Li, H., and Azarm, S., 2002, "An Approach for Product Line Design Selection Under Uncertainty and Competition," *ASME Journal of Mechanical Design*, 124(3), pp.385-392.

Meyer, M. H., Tertzakian, P., and Utterback, J. M., 1997, "Metrics for Managing Research and Development in the Context of the Product Family," *Management Science*, 43(1), pp. 88-111.

Michalek, J. J., Ceryan, O., Papalambros, P., and Koren, Y., 2005, "Manufacturing Investment and Allocation in Product Line Design Decision-Making," Proceedings of the ASME Design Engineering Technical Conference, DETC2005-84812, Long Beach, California, USA.

Pine, B. J. II, 1993, "Mass Customizing Products and Services," *Planning Review*, 21(4), pp. 6-8.

Raman, N., and Chhajed, D., 1995, "Simultaneous Determination of Product Attributes and Prices and Product Processes in Product-Line Design," *Journal of Operations Management*, 12(3), pp. 187-204.

Sawyer, J.F., 1970, *Line Balancing*, The Machinery Publishing, Brighton, United Kingdom.

Thomopoulos, N., 1970, "Mixed Model Line Balancing with Smoothed Station Assignments," *Management Science*, 16(9), pp. 993-603.

Thomopoulos, N., 1967, "Line Balancing-Sequencing for Mixed-Model Assembly," *Management Science*, 14(2), pp. B59-B75.

CHAPTER 4

ASSEMBLY SYSTEM RECONFIGURATION PLANNING

Abstract

Due to increased competition, the rate at which manufacturers introduce new product families to the market is increasing. However, the cost of changing manufacturing facilities to produce new products can outweigh the benefits obtained from increased revenues. Reconfigurable Manufacturing Systems (RMSs) have been proposed as a cost effective strategy for manufacturing product families. Although methods for measuring RMS scalability and convertibility exist, there is a lack of methods for obtaining reconfiguration plans for assembly systems. This chapter introduces assembly system reconfiguration planning (ASRP) as a method to obtain reconfiguration plans for assembly systems. Procedures are presented for solving the ASRP problem by dynamic programming and genetic algorithm. The computational accuracy and efficiency of the two solution approaches are compared.

4.1 Introduction

The decrease in the market life of products poses several challenges to manufacturers. Two of these challenges are reduced product development times and reduced product life cycles. Shorter product development times make it difficult to validate, test and build new processes and manufacturing systems to meet market needs in a timely fashion. Shorter production cycles mean that products no longer remain on the market for a sufficient amount of time to recover the initial cost of investment in expensive dedicated manufacturing systems.

Product families, which typically have modular architectures, have been used for efficiently evolving product designs over several generations [Seepersad *et al*, 2005, Martin and Ishii, 2002]. The modular architectures of product families allow for the reuse of product designs and manufacturing system capabilities. Reconfigurable manufacturing systems (RMS) have been proposed for the cost effective manufacture of product families [Koren *et al*, 1999]. By designing RMSs around product families, the manufacturing system can quickly and cost effectively respond to product family design changes. More recently, the co-evolution of product families and assembly systems has been proposed as a strategy for the concurrent development of several generations of product families and reconfigurable assembly systems [Bryan *et al*, 2007]. Co-evolution involves the concurrent design of product families and assembly systems within a single generation of the product family, and the reconfiguration of the assembly system between generations as the product family evolves.

This research focuses on the latter problem, the development of a strategy for reconfiguring assembly systems to meet the needs of evolving product families. More specifically, this chapter introduces a method for finding assembly system reconfiguration plans that minimize the life cycle cost of producing several generations of a product family. This approach is referred to as assembly system reconfiguration planning (ASRP).

Reconfiguration planning has recently received some attention by the research community. The research on reconfiguration planning had been focused on two main areas, scalability and convertibility. Scalability planning involves the design of manufacturing systems that can cost effectively change in capacity to respond to changes in product demand. Convertibility planning focuses on the design of manufacturing systems that respond to changes in product functionality. Procedures have been introduced for obtaining optimal plans for scalable and convertible machining systems [Spicer and Carlo, 2007, Ye and Liang, 2006, Son, 2000]. However, much less research has been done on reconfiguration planning for assembly systems.

Reconfiguration planning for automated assembly systems is essentially the same as that of machining systems. However, reconfiguration planning for semi-automated or manual assembly systems differs from machining systems as there is greater flexibility in the positioning of assembly tasks. This means that the search space for optimal reconfiguration plans is much larger and therefore the problem is more difficult. As a result, there is a need for the ASRP approach introduced in this chapter.

Although, strategies for multi-generational assembly system design exist, they do not necessarily minimize the life cycle cost of producing the product family. One such approach is to re-balance the assembly system for every generation of the product family [Gamberini *et al*, 2006]. This approach guarantees that the assembly system will operate at maximum efficiency within every generation. However, if reconfiguration costs are significant, this approach can prove to be expensive over the life cycle of the product family. Another approach is to balance the assembly system across several generations [Ko and Hu, 2007]. Although this approach may lead to operational inefficiencies within a given generation, it eliminates the need for reconfiguration. Therefore, if reconfiguration costs are significant, this approach may minimize life cycle costs.

The ASRP method proposed in this research does not presume a reconfiguration strategy. The model evaluates a wide space of assembly system configurations and then selects the reconfiguration plan that minimizes life cycle costs. Models for computing life cycle cost and optimization of the reconfigurations plans are introduced. The optimization model for the ASRP problem is combinatorial. This type of problem is usually difficult to solve exactly for moderately sized problems. Dynamic programming and genetic algorithm solution procedures are presented for finding solutions to the ASRP problem. The computational accuracy and efficiency of the two approaches are compared.

4.2 Representations

Models for the product family, assembly system, and life cycle costs used for ASRP are presented in the following sub-sections.

4.2.1 Product Family Representation

In typical assembly system design problems, a product is represented by a precedence diagram. *Precedence diagrams* are acyclic, directed graphs that illustrate the order in which assembly tasks are completed. A discussion on precedence diagrams is provided in [Becker and Scholl, 2006, Baybars, 1986].

Each product variant of a product family has a unique precedence diagram. A *product family precedence diagram* is a combined representation of the precedence diagrams of the product variants in the product family. The task-time in a product family precedence diagram is the volume-weighted-average task time across the product variants of the product family [Bryan *et al*, 2007]. As demand for the product variants in product family can change from one generation to another, the volume-weighted average task times for differentiating modules changes. The product family precedence diagram is assumed to be known a priori.

4.2.2 Assembly System Representation

An assembly system consists of serially arranged *workstations* that are linked together by a conveyor system [Baybars, 1986]. The product flows from one workstation to a next so that the set of tasks completed at a given workstation brings the product some steps closer to completion. In early assembly system design problems, duplication of tasks was not allowed. However, several researchers have realized the benefits of having parallel sets of tasks assigned at workstations [Askin, 1997, Pinto, 1991].

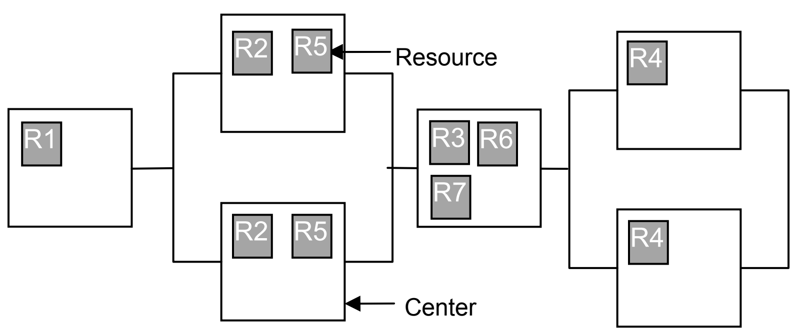


Figure 4-1 Assembly system representation

Figure 4-1 shows the model of the assembly system used in this research. *Centers* are the fixed infrastructure that is generic to all workstations. This includes, but not limited to, floor space and conveyors. *Resources* are the equipment required at a workstation to complete the tasks assigned to the workstation. Examples of task dependent resources are fixtures, pneumatic guns, and storage bins. The *layout* is the representation of the assembly system with just centers. The representation of the assembly system with both centers and resources is referred to as the *configuration* of the assembly system.

The duplicate sets of tasks that are assigned to a workstation are assumed to be completed at parallel centers. Hence, the assembly system layout has a *parallel-serial configuration*. A *pure parallel configuration* has one workstation and all required centers in parallel at that workstation. In this type of configuration, each worker completes all the tasks. In a *pure serial configuration*, each workstation has only one center. Each worker in a pure serial configuration completes a small set of tasks. Pure parallel and pure serial configurations are extremes of the parallel-serial configuration.

Since common tasks are assembled at the parallel centers at a given workstation, these parallel centers contain identical resources as shown in Fig. 4-1. Whereas, the consideration of resources may be ignored in the planning of single period assembly systems, the location of resources must be considered in multi-generation assembly system design problems such as ASRP. This is because the rearrangement of resources between product generations can have a significant impact on the reconfiguration cost and effort.

4.2.3 Assembly System Reconfiguration Planning

The main phases in the *life cycle* of a product family are initial development, production, and retirement. However, the term life cycle used in this chapter to only refers to the production phase of the life of the product family.

When a *design for product evolution* approach is taken, the product designs for each generation of the product family are known a priori. This knowledge is used to determine the best assembly system configuration for each generation. If the assembly system exists only to produce the given product family, the life cycle of the assembly system is equivalent to the life cycle of the product family it produces.

Figure 4-2 illustrates the life cycle of an assembly system which is composed of several generations. The total time in a single generation is the sum of the production and reconfiguration times. During reconfiguration, centers and resources are installed, removed, or rearranged and no products can be assembled. In traditional manufacturing systems, this reconfiguration time as well as the ramp down and ramp up periods before and after reconfiguration can be quite significant and must be accounted for. However when assembly systems are designed for product evolution, it is assumed that the equipment used is flexible and the reconfiguration time is a small fraction of the time production time. Although the reconfiguration time is small, the cost penalties incurred for this loss in production capacity during reconfiguration must still be accounted for. The periods of ramp down and ramp up before and after reconfiguration are assumed to be a very small fraction of the reconfiguration time and therefore negligible.

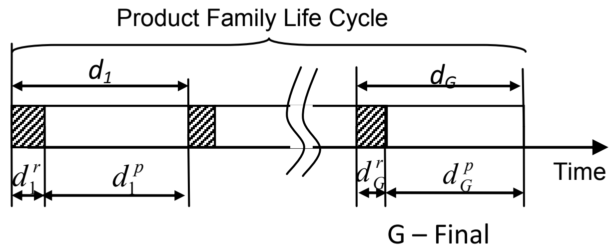


Figure 4-2 Product life cycle

In cases when no assembly system exists prior to the introduction of the first generation of the product family, reconfiguration during the first generation involves the installation of the initial assembly system. There is no need to consider the effect of lost production on life cycle costs for this case. This case is assumed for the ASRP approach. However, this assumption is not limiting and the ASRP model can easily be extended to include consideration of the existence of an assembly system prior to the introduction of the first generation product family.

4.3 Mathematical Model

Assembly system reconfiguration plans are selected according to their life cycle costs. Models for life cycle cost and optimization are presented in the following sub-sections. Symbols are defined in the Nomenclature section at the end of this chapter.

4.3.1 Life Cycle Costs

As shown in Fig. 4-2, each generation consists of a period of reconfiguration followed by a period of production. Therefore, the total cost incurred in each generation (θg) is the sum of the costs of production (η_g) and reconfiguration (ρ_g). Since ASRP occurs at the beginning of the first generation, the costs incurred in future generations must be discounted back to their present values. The net present life cycle cost for ASRP (Θ) is as defined in Eq. (4-1). This formulation assumes that θg is incurred at the beginning generation g.

$$\Theta = \sum_{g=1}^{G}\left[\frac{\theta_g}{npwf_g}\right] = \sum_{g=1}^{G}\left[\frac{\eta_g + \rho_g}{npwf_g}\right] \tag{4-1}$$

$npwf_g$ is the net present weight factor in generation g and is defined as follows:

$$npwf_g = (1+a)^{\sum_{g=1}^{G-1} d_g} \tag{4-2}$$

Several factors such as the cost of labor, utility, and inventory holding costs influence the variable cost of an assembly system. Only the labor costs are considered. Assuming that there is exactly one worker at every workstation, the variable cost in a given generation is computed as follows:

$$\eta_g = w_g d_g^p \sum_{m=1}^{M} N_{mg} \tag{4-3}$$

w_g and d_g^p are usually known parameters in assembly system design. As shown in Eq. (4-4), N_{mg}, the number of centers at workstation m in generation g depends on the tasks assigned to the workstation and the cycle time. The cycle time is a function of the production capacity needed to meet demand, Eq. (4-5). Equation (4-5) is substituted into Eq. (4-4) to obtain Eq. (4-6).

$$N_{mg} = \left\lceil \frac{\sum_{k=1}^{K} X_{kmg} t_{kg}}{ct_g} \right\rceil \tag{4-4}$$

$$ct_g = \frac{d_g^p}{v_g} \tag{4-5}$$

$$N_{mg} = \left\lceil \frac{v_g \sum_{k=1}^{K} X_{kmg} t_{kg}}{d_g^p} \right\rceil \tag{4-6}$$

Equation (4-7) shows that the cost of reconfiguration for a generation, ρ_g, is a function of the net equipment cost (ϕ_g), the cost for rearranging workstations (γ_g), and the cost for lost production during reconfiguration (ψ_g).

$$\rho_g = \phi_g + \gamma_g + \psi_g \tag{4-7}$$

Recall from Fig. 4-1 that the equipment at a workstation consists of centers, as well as resources. The net equipment cost, ϕ_g given in Eq. (4-8) is the net cost of purchasing and salvaging centers and resources in a given generation. The net cost of resources is dependent on the resource-workstation assignment, which is dependent on the task-workstation assignment. The indicator variable, Y_{img}, is used to represent this resource-workstation relationship. Its value is determined by using Eq. (4-9) and Eq. (4-10).

$$\phi_g = wpc_g \left[\sum_{m=1}^{M} \left(N_{mg} - N_{m,g-1} \right) \right]^+ \tag{4-8}$$

$$- wsc_g \left[\sum_{m=1}^{M} \left(N_{mg} - N_{m,g-1} \right) \right]^-$$

$$+ \sum_{i=1}^{I} wpr_{ig} \left[\sum_{m=1}^{M} \left(Y_{img} N_{mg} - Y_{im,g-1} N_{m,g-1} \right) \right]^+$$

$$- \sum_{i=1}^{I} wsr_{ig} \left[\sum_{m=1}^{M} \left(Y_{img} N_{mg} - Y_{im,g-1} N_{m,g-1} \right) \right]^-$$

$$\sum_{k=1}^{K} s_{ik} X_{kmg} \leq \Gamma Y_{img} \tag{4-9}$$

$$\sum_{k=1}^{K} s_{ik} X_{kmg} \geq Y_{img} \tag{4-10}$$

Γ is the upper bound on the affine function that defines the resource-workstation relationship. Hence, $\Gamma = K$, indicating that the upper bound occurs when all the tasks are assigned to workstation m and require resource i.

Rearrangement in a given generation involves installing, removing and retrofitting centers and resources at workstations in order to meet the production requirements for that generation. The work of retrofitting is considered to be analogous to adding and removing resources and is not given separate consideration. Rearrangement is typically performed manually. The cost of rearrangement in a given generation (γ_g) is computed as shown in Eq. (4-11).

$$\begin{aligned}\gamma_g = {} & w_g \ell ac \sum_{m=1}^{M} \left[N_{mg} - N_{m,g-1} \right]^{+} \\ & + w_g \ell rc \sum_{m=1}^{M} \left[N_{mg} - N_{m,g-1} \right]^{-} \\ & + w_g \sum_{i=1}^{I} \sum_{m=1}^{M} \ell ar_i \left[N_{mg} - N_{m,g-1} \right]^{+} \\ & + w_g \sum_{i=1}^{I} \sum_{m=1}^{M} \ell rr_i \left[N_{mg} - N_{m,g-1} \right]^{-}\end{aligned} \tag{4-11}$$

As indicated in Section 4.2.3, there is no place during reconfiguration. A manufacturer must therefore employ a strategy to make up for this lost production. Since demand is assumed to be met exactly in every generation, there is no inventory during reconfiguration to meet demand during the period of no production. Therefore, the strategy considered is the acceptance of the demand loss. It is noted that the g-1 product family is supplied during reconfiguration in generation g and the reconfiguration time is very small as compared to the production period. This leads to the assumption that the demand rate during reconfiguration in generation g is the same as the demand rate during generation g-1. This strategy and assumption lead to the cost for lost production (ψ_g) in Eq. (4-12).

$$\psi_g = b_g \frac{v_{g-1}}{d_{g-1}^{p}} \left\{ \begin{aligned} & \ell ac \sum_{m=1}^{M} \left[N_{mg} - N_{m,g-1} \right]^{+} \\ & + \ell rc \sum_{m-1}^{M} \left[N_{mg} - N_{m,g-1} \right]^{-} \\ & + \sum_{i=1}^{I} \sum_{m=1}^{M} \ell ar_i \left[Y_{img} N_{mg} - Y_{im,g-1} N_{m,g-1} \right]^{+} \\ & + \sum_{i=1}^{I} \sum_{m=1}^{M} \ell rr_i \left[Y_{img} N_{mg} - Y_{im,g-1} N_{m,g-1} \right]^{-} \end{aligned} \right\} \tag{4-12}$$

From Eq. (4-1) – Eq. (4-12), it is observed that θ_g varies linearly with the parameters for demand, reconfiguration time, labor rate and unit cost per product.

4.3.2 Optimization Formulation

The objective of the ASRP problem is to obtain the assembly system reconfiguration plan(s) that minimizes net present life cycle cost, Eq. (4-13). It does this by considering the entire space of possible assembly system reconfiguration plans. The optimization formulation for ASRP is given in Eq. (4-13)-Eq. (4-21).

$$\min_{\text{with respect to } N_{mg}, X_{kmg}} \quad \Theta = \sum_{g=1}^{G} \left\{ \frac{\eta_g}{npwf_g} + \frac{\rho_g}{npwf_g} \right\} \tag{4-13}$$

subject to

$$\sum_{m=1}^{M} X_{kmg} = z_{kg} \qquad \forall g \in \{1, \cdots, G\},\ k = \{1, \cdots, K\} \tag{4-14}$$

$$\sum_{k=1}^{K} s_{ik} X_{kmg} - \Gamma Y_{img} \le 0 \quad \forall i \in \{1, \cdots, I\}, g \in \{1, \cdots, G\} \tag{4-15}$$

$$Y_{img} - \sum_{k=1}^{K} s_{ik} X_{kmg} \le 0 \ \ \forall i \in \{1, \cdots, I\}, g \in \{1, \cdots, G\} \tag{4-16}$$

$$X_{kmg} - \sum_{n=1}^{m} X_{qng} \le 0 \ \ \forall g \in \{1, \cdots, G\}, k \in \{1, \cdots, K\}, m \in \{1, \cdots, M\}, q \in P(k) \tag{4-17}$$

$$N_{mg} - N_{m-1,g} \times \max(H_g, g = 1, \cdots, G) \le 0 \ \ \forall m \in \{1, \cdots, M\}, g \in \{1, \cdots, G\} \tag{4-18}$$

$$\sum_{k=1}^{K} X_{kmg} t_{kg} - ct_g N_{mg} \leq 0 \ \ \forall g \in \{1, \cdots, G\}, m \in \{1, \cdots, M\} \tag{4-19}$$

$$\sum_{m=1}^{M} N_{mg} - \max\left(H_g, g = 1, \cdots, G\right) \leq 0 \qquad g \in \{1, \cdots, G\} \tag{4-20}$$

$$N_{mg} \geq 0, \ X_{kmg} \in \{0,1\} \tag{4-21}$$

The objective function in Eq. (4-13) is different from the typical objective of assembly system design problems which are based on assembly line balancing (ALB) approaches. The objective of typical ALB is to maximize efficiency (minimize the number of centers), the system capacity only depends on task-workstation assignments. Therefore, the only decision variable needed is X_{kmg}. In order to allow for the consideration of a wider search space of assembly system design solutions in the ASRP approach, the assembly system capacity is decoupled from the task-workstation assignment. Therefore, the decision variables for the ASRP problem are N_{mg} and X_{kmg}.

The constraint in Eq. (4-14) guarantees that all tasks required in a given generation are assigned to a workstation. Equation (4-15) and Eq. (4-16) are the constraints that define the indicator variable Y_{img}. Precedence constraints between tasks are given in Eq. (4-17). Equation (4-18) ensures that centers are placed at workstations in an increasing order. The workstation capacity constraint in Eq. (4-19) ensures that sufficient centers are placed at a workstation to assemble the tasks assigned to the workstation. Equation (4-20) provides an upper bound on N_{mg} for all generations. This bound states that the total number of centers assigned in any generation must not exceed the number of centers required in the generation that requires the most number of centers to maximize efficiency. Feasibility constraints on the decision variables are given in Eq. (4-21).

4.4 Solution Approaches

The optimization formulation for finding minimal net present life cycle cost for ASRP is an integer, non-linear program (INLP). This type of problem is combinatorial and difficult to solve by exact optimization procedures. An exhaustive search of the entire space of assembly system reconfiguration plans will yield the optimal solution. However, this approach may not be practical for even moderately sized problems. For example, assume an ASRP problem with g generations and κ_g tasks in each generation g. Also assume that the NC is upper bound on N_{mg} for all g obtained from Eq. (4-20), i.e. NC=max(H_g, g=1,⋯,G). Then, Eq. (4-22) gives the upper bound on the number of possible assembly system reconfiguration plans ($NASRP$) when there are no precedence constraints between tasks. It can be seen that even for the case that precedence constraints lead to only one possible sequence, the value of $NASRP$ still increases exponentially with the number of tasks.

$$NASRP = \prod_{g=1}^{G}\left(\kappa_g! \times 2^{\kappa_g - 1} \times 2^{NC-1}\right) \tag{4-22}$$

As a result of the large solution space, more efficient methods than an exhaustive search is required for solving the ASRP problem. Dynamic programming and genetic algorithm procedures are introduced. Dynamic programming is guaranteed to find the optimal result. However, the size of the problem that can be solved is relatively small.

Genetic algorithm can be used to find solutions for larger problems. However, the optimality of the solutions cannot be guaranteed.

In order to develop a dynamic programming solution approach, the assembly system configurations for each generation must be determined. This requires the partitioning of sequences of tasks and centers into subsets. A new algorithm for grouping a sequence into subsets is developed.

The remainder of this section is organized as follows. First, an algorithm for grouping sequences into subsets is presented in Section 4.4.1. This is followed by a presentation of the dynamic programming and genetic algorithm in Sections 4.4.2, and 4.4.3 respectively.

4.4.1 Algorithm for Grouping Sequences into Subsets

A sequence of $\mathcal{D}$ items is represented as an ordered list of $\mathcal{D}$ blocks. Partitions are assumed to exist between every two blocks in the sequence. This representation of a sequence with partitions is shown in Fig. 4-3. It is observed that a sequence with $\mathcal{D}$ items will have at most $\mathcal{D}$-1 partitions between the items.

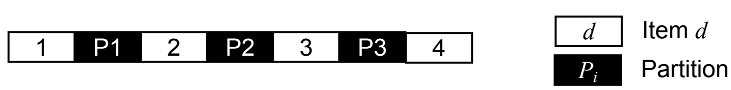


Figure 4-3 Representation of a sequence of items

$\{1,\cdots,p,\cdots,\mathcal{D}\text{-}1\}$ is the set of partitions between tasks. The grouping algorithm works by selecting some partitions from set $\mathcal{P}$ while masking the remaining partitions. The selected partitions indicate the points where the sequence is grouped into subsets.

The algorithm for finding all the ways that a sequence can be grouped into subsets is given in Fig. 4-4.

A sequence with D items is given. p is the number of partitions between items. The maximum value of p is D-1.

For i=0 to D-1
    Find S = the set of all combinations of D-1 partitions taken i at a time
    Compute $ns = \binom{D-1}{p} = |S|$
    For j=1 to ns
        Select the j^{th} combination from S
        Group the sequence into subsets using the partitions indicated by this j^{th} combination
        Store the grouped sequence
    End
End

Figure 4-4 Algorithm for generating all the subsets of a sequence

Each grouped sequence is written as $\{()()()\}$. The outer curly bracket represents the sequence while the inner brackets represent the groups formed from the sequence. Applying this representation to the sequence in Fig. 4-3, it can be seen that if no partitions are selected, the sequence will contain one group of tasks and is represented as $\{(1234)\}$. If partitions P1 and P3 are selected, the sequence contains three groups of tasks and is represented as $\{(1)(23)(4)\}$. The maximum number of grouped sequences that can be obtained from a sequence with $\mathcal{D}$ tasks is $\sum_{i=0}^{D-1}\binom{D-1}{i}$. However, these grouped sequences are not unique. Redundant grouped sequences are deleted.

4.4.2 Dynamic Programming

The ASRP problem is formulated as a deterministic, staged dynamic program. The stages in the network are the generations and the states are the possible assembly system configurations for each generation. The arc costs are functions of the variable costs (η_g) and reconfiguration costs (ρ_g) as given in Eq. (4-1). An optimal reconfiguration plan is obtained by finding the minimum cost path through the dynamic programming network. The recursive formulation for minimizing life cycle costs is given in Eq. (23).

$$f^*\left(con_g(j)\right)=\begin{cases} \min\left\{f^*\left(con_{g-1}(j)\right)+\theta_g\left(con_{g-1}(j'),con_g(j)\right)\right\} \text{ for } j'=1,\cdots,J_{g-1},\ g=1,\cdots,G \\ \min\left\{f^*\left(con_g(1)\right)\right\} \text{ for } g=G+1 \\ f\left(con_g(1)\right)=0 \text{ for } g=0 \end{cases} \tag{4-23}$$

In order to form the dynamic programming network, the states in the network must be determined. Each state is a possible assembly system configuration for each generation of the product family ($con_g(j)$). A new procedure for determining all the possible assembly system configurations in a given generation is introduced. Unlike existing line balancing procedures, the algorithm decomposes the grouping of tasks into subsets from the system layout.

The inputs to the assembly system configuration design problem are product configurations, assembly times and the demand for product. The following four steps are then used to determine the assembly system configurations.

1. Find all the possible assembly sequences.
2. Find all the possible subsets of tasks for each sequence.
3. Find all the possible assembly system layouts.

4. Map the subsets of tasks to assembly system layouts to obtain the assembly system configurations.

The first step of the procedure is the determination of all possible assembly sequences. There are two accepted methods for the generation of assembly sequences. One method uses the analysis of the product disassembly process to determine the relationship between the components in the product [Homem de Mello, 1991]. The other approach uses the answers to two questions in order to derive the relationships between components [De Fazio and Whitney, 1987]. The output of these two methods is the order in which components should be assembled, a.k.a. component sequences. Since an assembly process is simply the joining of components, the component sequences obtained from these analyses are considered as assembly task sequences.

Each sequence of tasks found in Step 1 is grouped into subsets in Step 2. It is necessary to maintain component sequences when grouping tasks into subsets. The sequence grouping algorithm introduced in Section 4.4.1 is used to group each sequence of tasks into subsets. Each grouping of a sequence is considered as a way of assigning the sequence of tasks to workstations.

The third step of the procedure is the determination of all the possible assembly system layouts. As defined in Section 4.2.2, a layout is the arrangement of centers without resources. The number of centers in each layout in each generation lies between an upper bound and a lower bound. The lower bound on the number of centers in generation g is H_g, the minimum number of centers required to complete the assembly of tasks in generation g. Traditional assembly line balancing approaches are used to determine H_g. The upper bound on the number of centers is the same for all generations. It is given by $max\{H_g, g=1,\ldots,G\}$.

The possible layouts for generation g are determined by considering the centers as an ordered list of items. The algorithm for grouping sequences into subsets presented in Section 4.4.1 is used to determine all the ways that sequences containing the number of centers between the upper and lower bound can be grouped into workstations. Each subset derived from a given sequence is a workstation while the centers in the subset are the parallel centers at the respective workstations. Equation (4-24) gives the computation of the available time at each workstation in the layout.

$$TA_{mg} = ct_g \times N_{mg} \tag{4-24}$$

The final step of the algorithm involves the mapping of each partitioned sequence of assembly tasks to each layout. When the subsets of assembly tasks are mapped to layouts, the resources required at the workstations in the layout are automatically assigned. Therefore, the mapping of partitioned sequences to assembly system layouts completes the process of obtaining assembly system configurations.

If the number of workstations in the assembly system layout is greater than the number of subsets of tasks, the partitioned sequence may be mapped to workstations in more than one way. On the other extreme, if the number of subsets of the task sequence is greater than the number of workstations, an infeasible mapping occurs. An infeasible mapping will also occur if the time available for assembly at any workstation is less than the time required to assemble the set of tasks assigned to the workstation i.e. if $TA_{mg} \leq \sum_{k=1}^{K} X_{kmg} t_{kg}$.

The following is an example to illustrate feasible and infeasible mappings. Consider the partitioned sequence of assembly tasks as {(T1,T3,T5),(T2,T4)} with the total assembly times of 15min and 8min for each subset respectively. The assembly system is assumed to have a cycle time of 10min. The layout shown in Fig. 4-5(a) is considered for assembly of the grouped sequence. Since the layout contains only workstation, the mapping between this assembly sequence and layout is infeasible. The mapping of the assembly sequence to the layout shown in Fig. 4-5(b) is feasible while the mapping to Fig. 4-5(c) is infeasible. The layouts in both Fig. 4-5(b) and Fig. 4-5(c) contain the right number of workstations. The layout in Fig. 4-5(b) has sufficient time available at workstation 1 for completing the first subset of assembly tasks while the layout in Fig. 4-5(c) does not. The mapping between the assembly sequence and the layout in Fig. 4-5(d) is also feasible. Figure 4-6 shows the assembly system configurations that can be obtained from the feasible assembly sequence layout mappings.

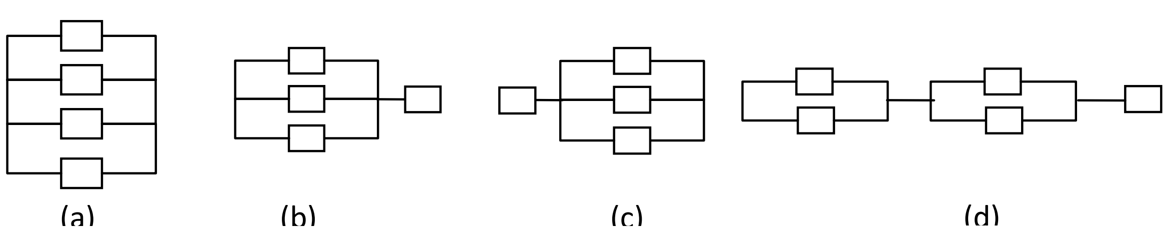


Figure 4-5 Examples of assembly system layouts

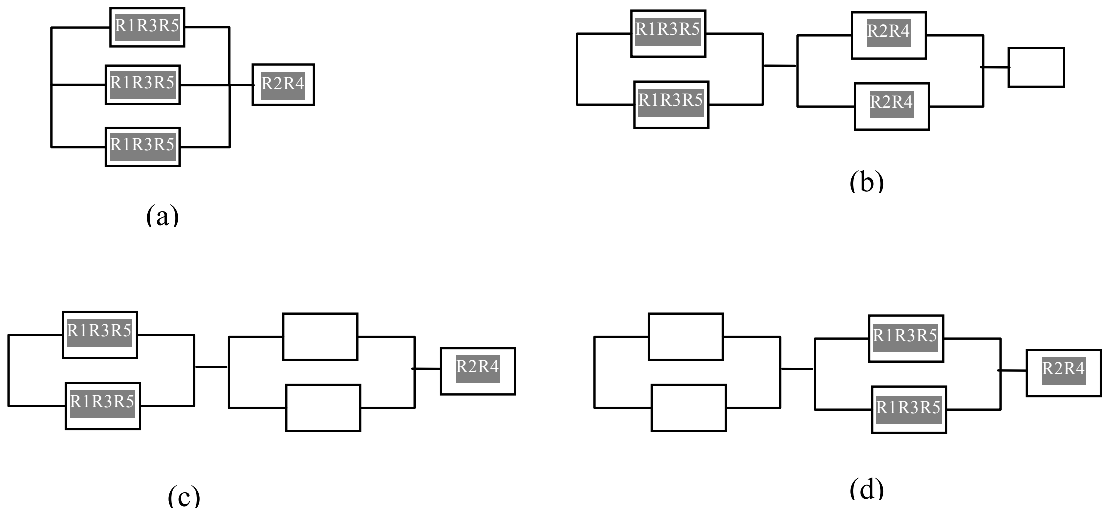


Figure 4-6 Examples of assembly system configurations

4.4.3 Genetic Algorithm

Genetic algorithms, which were first introduced by Holland (1975), have been shown to be computationally efficient for solving large combinatorial problems. The four main stages of formulating a GA are: (1) Representation of the solution; (2) Formation of an initial population; (3) Selection of members of the population for genetic manipulation; and (4) Genetic manipulation.

A real parameter, multi-sectioned chromosome string, Fig. 4-7, is used to represent the solution to the problem. The chromosome has two major sections corresponding to the two decision variables. The task-workstation assignment section corresponds to X_{kmg} and the center-workstation section corresponds to N_{mg}. Each of these sections is divided into sub-sections corresponding to the number of generations in the life cycle. In the task-workstation section, genes represent tasks while the values in the genes represent the workstations that the tasks are assigned to. The genes in the center-workstation sub-sections represent workstations and the values in the genes represent the number of centers at the given workstation. Applying these definitions to the chromosome representation in Fig. 4-7, it is observed that the chromosome represents an ASRP problem with two generations. It is also observed that the third task in the first generation is assigned to the second workstation and there are two parallel centers at this workstation.

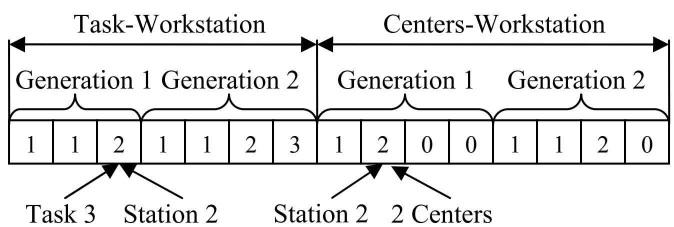


Figure 4-7 Chromosome representation

The initial population is generated randomly. In the task-workstation assignment section of the chromosome, a random number between one and the maximum number of possible workstations, H_g, is generated for the value of the gene. In the centers-workstation section, a random number between zero and *NC* is generated for the value of the gene. The maximum number of workstations and *NC* are determined by performing ALB for each of the generations before the start of the GA.

Selection is based on the fitness of the chromosomes. An elitist selection approach, which selects the chromosomes with the smallest fitness functions, is used. The percentage of the population that is selected is a parameter that is tuned in the genetic algorithm. A selection rate of 50% typically works well for this problem. The fitness function consists of the objective function shifted by the product of Lagrange multipliers and the constraints in Eq. (4-17 to 4-20). Using $r_e(N_{mg}, X_{kmg})$, e=1,…,4 to represent the constraints in Eq. (4-17 to 4-20) respectively and λ_e, e=1,…,4 to represent the Lagrange multipliers, the fitness function for selection is as defined in Eq. (4-25). Appropriate values for λ_e are determined by trial and error.

$$fitness = \sum_{g=1}^{G}\left\{\frac{\eta_g}{npwf_g}+\frac{\rho_g}{npwf_g}\right\}+\sum_{e=1}^{4}\lambda_e r_e\left(N_{mg}, X_{kmg}\right) \tag{4-25}$$

It is noted that the constraint in Eq. (4-14) and the feasibility constraints in Eq. (4-21) are automatically met by the chromosome representation. The constraints in Eq. (4-15) – Eq. (4-16) are determined from the given task-resource matrix.

The genetic operators used for solving the ASRP problem are crossover and mutation. A multi-point crossover approach is used. The crossover sites in both parents are first identified randomly and then the genes at these crossover sites are exchanged. A multi-point mutation approach is also used. The mutation sites in a chromosome are identified randomly. The values at these sites are then set to zero. The chromosome is repaired by randomly generating a feasible value for the mutated site using the same guidelines that were used in generating the initial population.

4.5 Examples

Two examples are developed in this section. The first example, which is small, is used to illustrate the implementation of the dynamic program and genetic algorithm for ASRP. The results and computational efficiencies of these two approaches are compared. The second example has more tasks and more generations than the first example. This example is solved by GA only as it is too large to be solved by dynamic programming.

The results of the ASRP problem are compared with the results of the conventional approach of rebalancing in every generation. The example is then extended to determine the effect of lowering the reconfiguration cost on the life cycle costs.

4.5.1 Example 1

Figure 4-8 is the product family precedence diagram for this example. The product family is produced for four generations. Tasks 1-2 are required in the first generation and one additional task is required in each succeeding generation. The product family assembly time (t_{kg}) of the tasks does not change in generations succeeding the first generation that the tasks are introduced. All tasks require a unique resource. The remaining parameters for this problem are provided in Table 4-1. There is one additional assumption that the resources for completing a task are not introduced in generations proceeding the first generation that the task introduced.

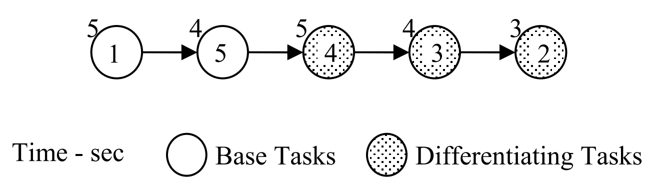


Figure 4-8 Precedence diagram for example 1

The dynamic program is implemented in Matlab. There is one feasible assembly sequence in all the generations. The number of states (assembly system configurations) in generation 1, 2, 3, and 4 are 30, 28, 24, and 16 respectively. This results in a total of 322560 possible solution paths. 60 of these paths lead to the optimal solution.

Table 4-1 Parameters for example 1

Parameter	Value
ℓac	28800 sec (1 day)
ℓrc	0.5* ℓac (0.5 days)
ℓar_i for all $i \in \{1,\ldots,5\}$	3600 sec (1 hr)
ℓrr_i for all $i \in \{1,\ldots,5\}$	0.5* ℓar_i (0.5 hr)
v_g for all $g \in \{1,\ldots,4\}$	1M units; $v_0 = 0$
d_g^P for all $g \in \{1,\ldots,4\}$	10M sec (1.4yrs); d_0^P $d_0^p = 0$
w_g for all $g \in \{1,\ldots,4\}$	$50/hr
b_g $g \in \{1,\ldots,4\}$	$5K/unit
wpc_g for all $g \in \{1,\ldots,4\}$	$10K
wsc_g for all $g \in \{1,\ldots,4\}$	$100
wpr_{ig} for all $g \in \{1,\ldots,4\}$, $i \in \{1,\ldots,5\}$	$1K
wpr_{ig} for all $g \in \{1,\ldots,4\}$, $i \in \{1,\ldots,5\}$	$10
a	0.1 (10%)

One optimal reconfiguration plan is shown in Fig. 4-9. The reconfiguration plan implements extra capacity as early as the first generation and then introduces resources in later generations as needed. The optimal net present life cycle cost is $5.3M. The net present variable and reconfiguration costs are $1.1M and $4.2M respectively. Figure 4-10 shows the net present reconfiguration and variable cost for each generation.

Figure 4-9 Optimal ASRP for example 1

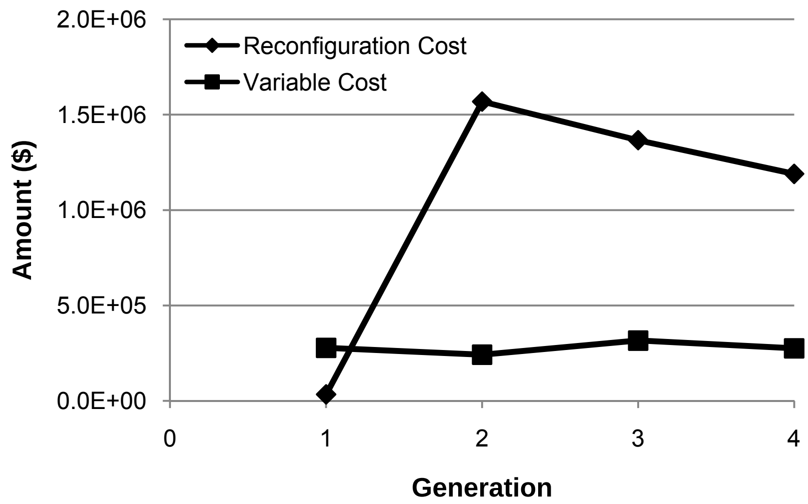


Figure 4-10 Net present costs for example 1

The GA was also implemented in Matlab. The parameters for solving the GA are provided in Table 4-2. The GA for the ASRP was able to find the same solution. The dynamic program and GA were solved in 0.4CPU sec and 56CPU sec respectively.

Table 4-2 GA parameters for example 1

GA Parameter	Value for Example 1
Population Size	500
λ_e for e=1,…,4	1000*wpc_g
Cross over rate	50%
Mutation Rate	10%
# of Generations	50

4.5.2 Example 2

The product family in this example is produced for six generations. The precedence diagram is shown in Fig. 4-11. There are 3, 5, 6, 8, 9, and 11 tasks in generations g=1,…,6 respectively. Tasks are introduced in increasing numerical order. As in Example 1, task times are unchanged in generations succeeding the first introduction that the tasks are introduced and each task requires a unique resource. The remaining parameters are the same as those given in Table 4-1 for example 1.

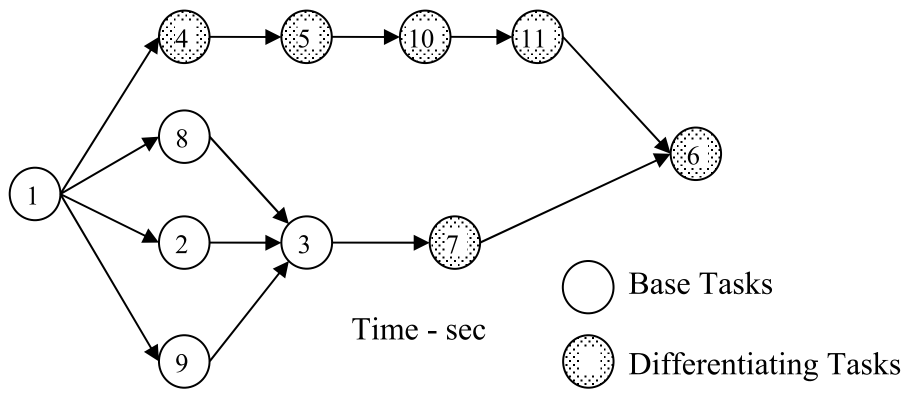


Figure 4-11 Precedence diagram for example 2 [Jackson, 1956]

The results for dynamic programming could not be obtained because the state space was too large to implement the DP in Matlab. However, as shown in example 1, the GA finds good solutions to the ASRP problem. The GA parameters are provided in Table 4-3.

Table 4-3 GA parameters for example 2

GA Parameter	Value for Example 2
Population Size	4000
λ_e for e=1,…,4	100 000*wpc_g
Cross over rate	50%
Mutation Rate	10%
# of Generations	100

The net present life cycle cost, net present variable cost and net present reconfiguration cost are \$12.7M, \$2.9M, and \$9.8M respectively. The task-workstation assignments are shown in Table 4-4.

Table 4-4 Example 2 – task-workstation assignments

	Gen1	Gen2	Gen3	Gen4	Gen5	Gen6
Workstation 1	1	1	1	1	1	1
Workstation 2	2	2,4	2,4	2,4	2,4	2,4
Workstation 3				8	8,9	8,9
Workstation 4		5	5	5	5	5,10
Workstation 5	3	3	3	3	3	3,11
Workstation 6			6	6,7	6,7	6,7

This example is also solved by the traditional approach in which the assembly system is rebalanced in every generation. The net present life cycle cost is \$51.4M, net present reconfiguration cost is \$49.5M, and net present variable cost is \$1.9M. Fig. 4-12 is a comparison of the ASRP to rebalancing in every generation.

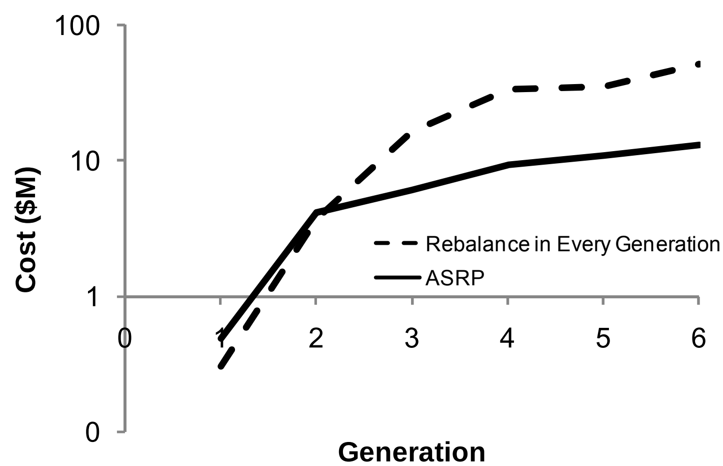


Figure 4-12 Net present life cycle cost for ASRP and rebalancing in every generation

The example is then resolved with the ASRP approach using the same parameters with the exception of the following: $\ell ac=\ell ar_i$ for i=1,…,11=10sec. and $\ell rc=\ell rr_i$ for i=1,…,11=5sec, b_g=\$100. Therefore, the reconfiguration costs are much lower. The net present life cycle cost, net present variable cost and net present reconfiguration cost for example 2 with these new parameters are \$2.0M, \$1.9M, \$60K respectively. As with the original parameters, six workstations with one task per workstation are required in every generation. Tables 4-5 show the workstation utilization for this new scenario.

Table 4-5 Example 2 with lower reconfiguration costs – task-workstation assignments

	Gen1	Gen2	Gen3	Gen4	Gen5	Gen6
Workstation 1	1	1,4,5	1,4,5	1,4,5	1,4,5	1,4,5
Workstation 2				8	8,9	8,9
Workstation 3						2
Workstation 4	2,3	2,3	2,3	2,3	2,3	3,10
Workstation 5						11
Workstation 6			6	6,7	6,7	6,7

4.5.3 Discussion

The solution space of possible configurations included pure parallel, pure serial, and all the parallel-serial assembly systems between these extremes. From the results to Examples 1 and 2, it is observed that pure serial line configurations are selected. This result could be due to the fact that the task times were less than the cycle time in these examples. However, this observation requires further investigation before a conclusion can be made.

It is also observed that the selected assembly system reconfiguration plans favored the installation of extra capacity in the first generation, even when reconfiguration costs are much less than variable costs. Recall that there is no cost for lost production in the first generation. Therefore, the results show that savings are gained by implementing extra capacity in the first generation and avoiding the cost of lost production between generations.

Example 2 shows that although it may sometimes be initially be more beneficial to rebalance the assembly system in every generation, over the life cycle, the ASRP approach leads to lower costs as shown in Fig. 4-12. This is because rebalancing in every generation ensures that the assembly system operates efficiently at minimum cost within a generation. However, the reconfiguration costs of getting from one low cost state to another may be high. The ASRP approach may not be efficient within every generation. However, it finds the minimum cost over the life cycle.

From the results of example 2, it is also observed that when the cost of reconfiguration is much higher than variable costs, a strategy that minimizes the number of times reconfiguration occurs is preferred. This strategy is not operationally efficient within each generation, but minimizes the total life cycle costs. On the other hand, when variable costs are more significant than reconfiguration costs, then a strategy that is operationally efficient in every generation is preferred. Note that although this solution may lead to the same number of workstations and group tasks in the same way as an assembly line re-balancing problem, the number of workstations in the assembly system and the workstation to which tasks are assigned are different. This is due to the extra capacity that is installed in the line to allow for the introduction of future tasks.

4.6 Conclusions

Assembly system reconfiguration planning (ASRP) was introduced as an approach for finding assembly system reconfiguration plans that minimizes the net present life cycle cost for producing an evolving product family. By decoupling task-workstation assignments from the determination of the number of centers required at every station, the ASRP approach was able to consider the entire set of possible assembly system reconfiguration plans.

Dynamic programming and genetic algorithm approaches for solving the ASRP problem were introduced. In order to generate the state space for the dynamic program, a new algorithm for grouping sequences into subsets was developed. For a small example, the same result was obtained with both the dynamic program and the genetic algorithm approaches. However, the state space was too large to solve a larger example by the dynamic programming approach.

The results of examples indicate that the installation of extra capacity in early generations minimizes the life cycle cost of producing the product family. The results also show that serial line configurations are preferred when task times are smaller than the cycle time.

Acknowledgements

The author would to acknowledge co-authorship of this paper with Prof. S. Jack Hu and Prof. Yoram Koren. The author would like to acknowledge the support of the Engineering Research Center for Reconfigurable Manufacturing Systems at the University of Michigan (NSF Grant No. EEC95-92125). In addition, we would like to thank Jeff Abell from the General Motors for his numerous contributions.

Nomenclature

Sets

$\{1,\ldots d,\ldots,\mathcal{D}\}$	Set of items in a sequence
$\{1,\ldots g,\ldots,G\}$	Set of generations
$\{1,\ldots i,\ldots,I\}$	Set of resources
$(1,\ldots k,\ldots,K\}$	Set of tasks
$\{1,\ldots m,\ldots,M\}$	Set of workstations
$\{1,\ldots p,\ldots,\mathcal{D}\text{-}1\}$	Set of partitions

Variables

Nmg	Number of centers at workstation m in generation g
Xkmg	1 if task k is assigned to workstation m in generation g, 0 otherwise
Yimg	1 if i task is assigned to workstation m in generation g, 0 otherwise

Lower Case Letters

a	Annual discount rate of money
b_g	Cost per unit that the manufacturer pays for product to make up for lost production in generation g
$con_g(j)$	State in the dynamic programming network. Configuration j in generation g.
ct_g	Cycle time in generation g
d_g	Duration of generation g
d_g^p	Duration of production in generation g
d_g^r	Duration of reconfiguration in generation g
$f^*(con_g(j))$	The minimum cost reconfiguration plan for all generations up to g where j is the configuration being considered.
ℓac	Time it takes one worker to install one center
ℓar_i	Time it takes one worker to install one unit of resource i
ℓrc	Time it takes one worker to remove one center
ℓrr_i	Time it takes one worker to remove one unit of resource i
$npwf_g$	Net present weight factor in generation g
ns	The number of elements in set $\mathcal{S}$
r_e	Constraint equations
s_{ik}	Task-resource parameter. 1 if task k requires resource i, 0 otherwise
t_{kg}	Time required to complete task k in generation g
w_g	Labor cost per unit time
wpc_g	Purchase cost per unit center in generation g
wpr_{ig}	Purchase cost per unit resource i in generation g
wsc_g	Salvage value per unit center in generation g

wsr_{ig}	Salvage value per unit resource i in generation g
v_g	Demand (volume of production) in generation g
z_{kg}	1 if task k is used in generation g, 0 otherwise

Upper Case Letters (Non-Variables)

D	The number of items in a sequence
H_g	Number of workstations resulting from the assembly line balancing solution in generation g
NC	Upper bound on the number of centers
$P(k)$	Set of predecessors of task k
S	The set of all combinations of x items taken y at a time
TA_{mg}	Available time for assembly at workstation m in generation g

Greek Letters

γ_g	The cost for rearranging workstations in generation g
η_g	Labor rate in generation g
θ_g	Cost of production in generation g
κ_g	Number of tasks required in generation g
λ_e	Lagrange multiplier e
ρ_g	Reconfiguration costs incurred in generation g
ϕ_g	Net equipment cost in generation g
ψ_g	Cost for lost production in generation g
Γ	Upper bound on the affine function that defines the resource-workstation relationship
Θ	Net present life cycle cost

Other Symbols

$\lceil x \rceil$	x rounded to the next nearest integer
$[x]^+$	The output of this operator is $\lvert x \rvert$ if $x \geq 0$, 0 otherwise
$[x]^-$	The output of this operator is $\lvert x \rvert$ if $x \leq$, 0 otherwise

References

Askin, R., G., and Zhou, M., 1997, "A Parallel Station Heursitic for the Mixed-Model Production Line Balancing Problem," *International Journal of Production Research*, 35(11), pp. 3095-3105.

Baybars, I., 1986, "A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem," *Management Science*, 32(8), pp. 909-932.

Becker, C., and Scholl, A., 2006, "A Survey on Problems and Methods in Generalized Assembly Line Balancing," *European Journal of Operational Research*, 168(3), pp. 694-715.

Bryan, A., Ko, J., Hu, S. J., and Koren, Y., 2007, "Co-Evolution of Product Families and Assembly Systems," *CIRP Annals*, 56(1), pp. 21-44.

Bryan, A., Hu, S. J., and Koren, Y., 2007, "Concurrent Design of Product Families and Assembly Systems," Proceedings of the ASME Int. Manuf. Sci. and Eng. Conf., Atlanta, GA, USA.

De Fazio, T. L., and Whitney, D. E., 1987, "Simplified Generation of All Mechanical Assembly Sequences," *IEEE Journal of Robotics and Automation*, 3(6), pp. 640-657.

Gamberini, R., Grassi, A., and Rimini, B., 2006, "A New Multi-Objective Heuristic Algorithm for Solving the Stochastic Assembly Line Re-Balancing Problem," *International Journal of Production Economics*, 102(2), pp. 226-243.

Holland, J. H., 1975, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, Michigan, USA.

Homem de Mello, L. S, 1991, "A Correct and Complete Algorithm for the Generation of Mechanical Assembly Sequences," *IEEE Journal of Robotics and Automation*, 17(2), pp. 228-240.

Jackson, J., 1956, "A Computing Procedure for Assembly Line Balancing," *Management Science*, 2(3), pp. 261-271.

Ko, J., and Hu, S. J., 2007, "Manufacturing System Design Considering Generation Product Evolution and Task Recurrence," Proceedings of the ASME Int. Manuf. Sci. and Eng. Conf., Atlanta, GA, USA.

Koren, Y., Heisel, U., Jovan, F., Moriwaki, T., Pritschow, G., Ulsoy, G., and Van Brussel, H., 1999, "Reconfigurable Manufacturing Systems," *CIRP Annals*, 48(2), pp. 527-540.

Martin, M. V., and Ishii, K., 2002, "Design for Variety: Developing Standardized and Modularized Product Platform Architectures," *Research in Engineering Design*, 13(4), pp. 213-235.

Pinto, P., Dannenbring, D. G., and Khumawala, B. M., 1981, "Branch and Bound and Heursitic Procedures for Assembly Line Balancing with Paralleling of Stations," *International Journal of Production Research*, 19(5), pp. 565-576.

Seepersad, C. C., Mistree, F., and Allen, J. K, 2005, "Designing Evolving Families of Products Using the Utility-Based Compromise Decision Support Problem," *International Journal of Mass Customisation*, 1(1), pp. 37-64.

Son, S. Y., 2000, *Design Principles and Methodologies for Reconfigurable Machining Systems*, Ph.D. thesis, University of Michigan, Ann Arbor, MI.

Spicer, P., and Carlo, H., 2007, "Integrating Reconfiguration Cost Into the Design of Multi-Period Scalable Manufacturing Systems," *ASME Journal of Manufacturing Science and Engineering*, 129(1), pp. 202-210.

Ye, H., and Liang, M., 2006, "Simultaneous Modular Product Scheduling and Manufacturing Cell Reconfiguration Using a Genetic Algorithm," *ASME Journal of Manufacturing Science and Engineering*, 128(4), pp. 984-995.

CHAPTER 5

SUMMARY AND FUTURE RESEARCH

5.1 Summary

In this dissertation, the *co-evolution of product families and assembly systems* was introduced as a new method for jointly evolving product families and reconfiguring assembly systems over several generations. The formal definition of co-evolution was given as follows: "Co-evolution of product families and assembly systems is a method the joint design and reconfiguration of the functionality and capacity of product families and their corresponding assembly systems within and across product generations in order to meet present and future product family and assembly system needs." It was observed that co-evolution differs from existing concurrent engineering methods as the co-evolution methodology plans for future changes in the very first generation while concurrent engineering methods plan for each generation individually. The two main phases of the co-evolution methodology were given as the concurrent design of the first generation of the product family and the assembly system and the reconfiguration of the assembly system between product generations. Furthermore, the mathematical models necessary for the implementation of the co-evolution method were identified as models for the concurrent design of the first generation of the product family and assembly system, models for the evolution of the product family and models for the reconfiguration of the assembly system. In this dissertation, mathematical models and solution procedures for the concurrent design of the product family and assembly system, and the reconfiguration of the assembly system were presented.

In Chapter 2, a procedure was presented for concurrently designing the product family and assembly system. The design problem was formulated as a multi-objective optimization problem in which the oversupply of product modules was minimized and the efficiency of the assembly system was maximized. The formulation developed was an integer non-linear program (INLP). It was noted that these problems are difficult to solve by exact optimization procedures. A genetic algorithm formulation was presented for solving the problem. Case studies were used to show that the product variants selected for the product family and their corresponding mix ratios does have an impact on the design of the assembly system.

In Chapter 3, the problem of concurrently designing the first generation of the product family and assembly system was reformulated as a single objective, profit maximization problem. The profit maximization formulation consisted of two sub-objectives, the maximization of revenues and the minimization of costs. These revenues and costs were dependent on the demand for the product variants in the product family. Consumer choice modeling was used to estimate the volumes of the product variants that should be produced in the product family. This concurrent design problem was also formulated as an INLP and a genetic algorithm was presented for solving the problem. Analysis of case studies indicated that the profit maximizing product family did not necessarily have the highest revenue or capture the greatest market share. However, it did find the solution with the highest profit overall. The case studies also showed that this approach is better than the existing sequential approach which first finds the product family that maximizes revenue and then the assembly system that minimizes cost. Only when the profit maximizing product family was able to be assembled for the same cost of the minimum cost assembly system, (i.e. there is no difference in assembly costs among all the product variants) that the sequential approach finds the same solution as the concurrent approach.

Chapter 4 introduced the assembly system reconfiguration planning (ASRP) problem for designing several generations of the assembly system that minimizes the life cycle cost of assembly for several known generations of the product family. The life cycle cost consisted of both the variable cost of operating the assembly system within a generation and the cost of reconfiguring the assembly system between product generations. The formulation for the ASRP problem was an INLP. A dynamic programming formulation was introduced for finding optimal solutions to the ASRP problem. In order to find the states for dynamic programming, a method was developed for generating the entire set of possible assembly system configurations. Since dynamic programming is intractable for large problems, a genetic algorithm was also introduced for solution of the ASRP problem. It was shown that the genetic algorithm found the same optimal solution as the dynamic program for small case studies. The case studies indicated that over the life cycle of the product family, the sequence of configurations obtained from the ASRP problem had a lower life cycle cost than the generation by generation approach to assembly system planning. This was because the ASRP problem built in extra capacity into the system in the very first generation to account for future changes. When the reconfiguration costs were much more significant than the variable costs, solutions that limit the number of assembly system reconfigurations were favored by the ASRP approach. The reverse was true when the reconfiguration costs were relatively low.

5.2 Original Contributions

The following are the original contributions in this dissertation:

1. Co-evolution of product families and assembly systems was introduced as a new concept for the joint evolution of several generations of product families and assembly systems.
2. A mathematical formulation and a solution procedure was introduced for solving the problem of the concurrent design of a product family and assembly system from an operations perspective. The methods introduced not only determined the product variants that should be in the product family, it also determined the explicit design of the assembly system.
3. A mathematical formulation and solution procedure was introduced for solving the problem of the concurrent design of a product family and assembly system from a combined operations and marketing perspective. In addition to obtaining the product family and the explicit design of the assembly system, estimates of the market share, and the relative revenue and costs of production are obtained from solving this problem.
4. The ASRP problem is introduced as a means for designing several generations of assembly systems that minimizes the life cycle cost of an evolving product family. Two approaches to solving the ASRP problem are also introduced.

5. A method for obtaining all the possible parallel-serial assembly system configurations for producing a known product family is introduced.

5.3 Suggested Future Research

The models and representations introduced in this dissertation provide a framework for making concurrent design decisions from a quantitative perspective. The true value of these models does not lie in the absolute numerical results obtained from solving the problems, but in the understanding they bring about how marketing, design, and manufacturing variables trade-off to affect product family and assembly system design. Therefore, these models are a good tool to help engineers and managers in decision making. There are several possible extensions to this dissertation. The following outlines a few of them.

1. Stochastic Considerations

The models introduced in this dissertation assumed that the input parameters, such as module assembly times, were deterministic. These models are useful for understanding the interactions between the design of the product family and assembly system. However, the manufacturing environment is dynamic. A valuable extension of this work would be to make some of the input parameters stochastic.

2. Product Module Design

The product family design problems introduced in this dissertation involved the selection of modules for product variants and product variants for the product family. It was assumed that the modules themselves were already designed, i.e. the alternatives for each module and the parameters for each module alternative were known a priori. Since the module parameters can affect both consumer decisions and assembly costs, it would be useful to make the explicit design of modules part of the decision making in future problems.

3. Incorporation of Product Module Sequencing

Throughout this dissertation, the precedence relations among product modules were represented by precedence diagrams. These precedence diagrams were assumed to be known. Precedence diagrams are artifacts that give a mathematical representation to the assembly order of modules. Therefore, the use of precedence diagrams strongly influences the assignment of tasks to workstations and hence the final design of the assembly system. However, a single precedence diagram does not necessarily capture all the possible assembly sequences. By the development of methods to mathematically represent the feasible set of all possible assembly sequences and assembly system configurations, the solution space could be increased. This extension can lead to improved solutions for assembly system design.

4. Alternative Objective Functions

The last two problems introduced in this dissertation, considered cost as the objective function. There are several other factors that affect decision making in a manufacturing company. These other considerations can be added to the existing objectives or used instead of the existing objectives to assist in decision making. For example, the concurrent design of the product family and assembly system can be pursued to determine the right type and quantity of product variety that minimizes lead time or complexity. The ASRP can also be formulated to maximize the end of life reuse and/or the life cycle cost of the product family.

5. Concurrent product family evolution and assembly system reconfiguration

In the ASRP problem, the future generations of the product family were assumed to be known. Therefore, the minimum life cycle cost of assembly applies to just one product family. This problem can be extended to incorporate design of the product family as well.

6. Alternative Assembly System Configurations

Throughout this dissertation, the assembly system was assumed to be a mixed model assembly line with parallel-serial configurations. Although this is the most common assembly system configuration for manual assembly, other assembly system configurations are possible. An interesting extension of this work would be to make the assembly line configuration a variable. This may lead to solutions with much lower assembly system costs. For example, if the product family has two product variants with large differences in assembly times but approximately the same volumes, it may be more efficient and hence cheaper to produce each product variant on its own assembly line rather than on a mixed model assembly line.

7. Equipment Selection for the ASRP Problem

The ASRP problem assumed that only one type of equipment was available for assembly of each individual product module. Another important decision that assembly system designers are faced with is determination of the most appropriate type of equipment that should be used in the assembly system. The cost of equipment and assembly efficiency often affects these decisions. The existing ASRP problem can be extended to include the selection of equipment as a decision variable.

BIBLIOGRAPHY

Abdi, M. R., and Labib, A. W., 2004, "Grouping and Selecting Products: The Design Key of Reconfigurable Manufacturing Systems," *International Journal of Production Research*, 42(3), pp. 521-546.

Askin, R., G., and Zhou, M., 1997, "A Parallel Station Heursitic for the Mixed-Model Production Line Balancing Problem," *International Journal of Production Research*, 35(11), pp. 3095-3105.

Balakrishnan, P. V., and Jacob, V. S., 1996, "Genetic Algorithms for Product Design," *Management Science*, 42(8)pp. 1105-1117.

Baybars, I., 1986, "A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem," *Management Science*, 32(8), pp. 909-932.

Becker, C., and Scholl, A., 2006, "A Survey on Problems and Methods in Generalized Assembly Line Balancing," *European Journal of Operational Research*, 168(3), pp. 694-715.

Boothroyd, G., and Dewhurst, P., 1989, *Product Design for Assembly*, Wakefield, RI.

Bryan, A., Hu, S. J., and Koren, Y., 2007, "Concurrent Design of Product Families and Assembly Systems," Proceedings of the ASME Int. Manuf. Sci. and Eng. Conf., Atlanta, GA, USA.

Bryan, A., Ko, J., Hu, S. J., and Koren, Y., 2007, "Co-Evolution of Product Families and Assembly Systems," *CIRP Annals*, 56(1), pp. 21-44.

Bryan, A., Hu, S. J., Koren, Y., 2007, "Concurrent Product Portfolio Planning and Mixed Product Assembly Line Balancing," *Chinese Journal of Mechanical Engineering*, 20(1), pp. 96-99.

Child, P., Diederichs, R., and Sanders, F. H., 1991, "The Management of Complexity," *Sloan Management Review*, 33(1), pp. 73-80.

De Fazio, T. L., and Whitney, D. E., 1987, "Simplified Generation of All Mechanical Assembly Sequences," *IEEE Journal of Robotics and Automation*, 3(6), pp. 640-657.

Fellini, R., Kokkolaras, M., Papalambros, P., and Perez-Duarte, A., 2005, "Platform Selection Under Performance Bounds in Optimal Design of Product Families," *ASME Journal of Mechanical Design*, 37(4), pp. 524-535.

Frei, R., Ribeiro, L., and Barata, J., 2007, "Evolvable Assembly Systems: Towards User Friendly Manufacturing," Proceedings of the IEEE International Symposium on Assembly and Manufacturing, Ann Arbor, Michigan, USA.

Gamberini, R., Grassi, A., and Rimini, B., 2006, "A New Multi-Objective Heuristic Algorithm for Solving the Stochastic Assembly Line Re-Balancing Problem," *International Journal of Production Economics*, 102(2), pp. 226-243.

Garey, M. R., and Johnson, D. S., 1979, *Computers and Intractability: A Guide to NP-Completeness*, W. H. Freeman, San Francisco, California, USA.

Gonzalez-Zugasti, J., Otto, K., and Baker, J., 2000, "A Method for Architecting Product Platforms," *Research in Engineering Design*, 12(2), pp.61-72.

Green, P., and Krieger, A., 1985, "Models and Heuristics for Product Line Selection," *Marketing Science*, 4(1), pp.1-19.

Green, P. E., and Krieger, A. M., 1987, "Consumer Based Approach to Designing Product Line Extensions," *Journal of Product Innovation Management*, 4(1), pp. 21-32.

Green, P., and Krieger, A., 1989, "Recent Contributions to Optimal Product Positioning and Buyer Segmentation," *European Journal of Operations Research*, 41(2), pp.127-141.

Green, P. E., and Srinivasan, V., 1990, "Conjoint Analysis in Marketing: New Developments with Implications for Research and Practice," *Journal in Marketing*, 54(4), pp. 3-19.

Hernandez, G., Allen, J., Simpson, T., Bascaran, E., Avila, L., and Salina, F., 2001, "Robust Design of Families of Products with Production Modeling and Evaluation," *ASME Journal of Mechanical Design*, 123(2), pp.183-190.

Holland, J. H., 1975, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, Michigan, USA.

Homem de Mello, L. S, 1991, "A Correct and Complete Algorithm for the Generation of Mechanical Assembly Sequences," *IEEE Journal of Robotics and Automation*, 17(2), pp. 228-240.

Jackson, J., 1956, "A Computing Procedure for Assembly Line Balancing," *Management Science*, 2(3), pp. 261-271.

Jiao, J., Simpson, T. W., and Siddique, Z., 2006, "Product Family Design and Product-Based Platform Development: A State-of-the-Art Review," *Journal of Intelligent Manufacturing*, Special Issue, pp. 1-36.

Jiao, J., and Tseng, M. M., 2000, "Fundamentals of Product Family Architecture," *Integrated Manufacturing Systems*, 11(7), pp. 469-483.

Jiao, J., and Zhang, Y, 2005, "Product Portfolio Planning with Customer-Engineering Interactions," *IIE Transactions*, 37(9), pp. 801-814.

Kaul, A., and Rao, V. R., 1995, "Research For Product Positioning and Design Decisions: An Integrative Review," *Research in Marketing*, 12(4), pp. 293-320.

Ko, J., and Hu, S. J., 2007, "Manufacturing System Design Considering Generation Product Evolution and Task Recurrence," Proceedings of the ASME Int. Manuf. Sci. and Eng. Conf., Atlanta, GA, USA.

Koren, Y., Heisel, U., Jovan, F., Moriwaki, T., Pritschow, G., Ulsoy, G., and Van Brussel, H., 1999, "Reconfigurable Manufacturing Systems," *CIRP Annals*, 48(2), pp. 527-540.

Krishnan, V., and Ulrich, K. T., 2001, "Product Development Decisions: A Review of the Literature," *Management Science*, 47(1), pp. 1-21.

Leu, Y., Matheson, L., and Rees, L., 1994, "Assembly Line Balancing Using Genetic Algorithms with Heuristic Generation Initial Population and Multiple Evaluation Criteria," *Decision Sciences*, 25(4), pp. 581-606.

Li, H., and Azarm, S., 2002, "An Approach for Product Line Design Selection Under Uncertainty and Competition," *ASME Journal of Mechanical Design*, 124(3), pp.385-392.

Martin, M. V., and Ishii, K., 2002, "Design for Variety: Developing Standardized and Modularized Product Platform Architectures," *Research in Engineering Design*, 13(4), pp. 213-235.

Meyer, M. H., Tertzakian, P., and Utterback, J. M., 1997, "Metrics for Managing Research and Development in the Context of the Product Family," *Management Science*, 43(1), pp. 88-111.

Michalek, J. J., Feinberg, F. M., and Papalambros, P. Y., 2005, "Linking Marketing and Engineering Product Design Decisions via Analytical Target Cascading," *Journal of Product Innovation Management*, 22(1), pp. 42-62.

Michalek, J. J., Ceryan, O., Papalambros, P., and Koren, Y., 2005, "Manufacturing Investment and Allocation in Product Line Design Decision-Making," Proceedings of the ASME Design Engineering Technical Conference, DETC2005-84812, Long Beach, California, USA.

Pine, B. J. II, 1993, “Mass Customizing Products and Services,” *Planning Review*, 21(4), pp. 6-8.

Pinto, P., Dannenbring, D. G., and Khumawala, B. M., 1981, “Branch and Bound and Heursitic Procedures for Assembly Line Balancing with Paralleling of Stations,” *International Journal of Production Research*, 19(5), pp. 565-576.

Raman, N., and Chhajed, D., 1995, “Simultaneous Determination of Product Attributes and Prices and Product Processes in Product-Line Design,” *Journal of Operations Management*, 12(3), pp. 187-204.

Rekiek, B., De Lit, P., and Delchambre, A., 2000, “Designing Mixed Product Assembly Lines,” *IEEE Journal of Robotics and Automation*, 16(3), pp. 268-280.

Sawyer, J.F., 1970, *Line Balancing*, The Machinery Publishing, Brighton, United Kingdom.

Seepersad, C. C., Mistree, F., and Allen, J. K, 2005, “Designing Evolving Families of Products Using the Utility-Based Compromise Decision Support Problem,” *International Journal of Mass Customisation*, 1(1), pp. 37-64.

Son, S. Y., 2000, *Design Principles and Methodologies for Reconfigurable Machining Systems*, Ph.D. thesis, University of Michigan, Ann Arbor, MI.

Spicer, P., and Carlo, H., 2007, “Integrating Reconfiguration Cost Into the Design of Multi-Period Scalable Manufacturing Systems,” *ASME Journal of Manufacturing Science and Engineering*, 129(1), pp. 202-210.

Thomopoulos, N., 1970, “Mixed Model Line Balancing with Smoothed Station Assignments,” *Management Science*, 16(9), pp. 993-603.

Thomopoulos, N., 1967, “Line Balancing-Sequencing for Mixed-Model Assembly,” *Management Science*, 14(2), pp. B59-B75.

Travaini, E., Pedrazzoli, O., Rinaldi, R., and Boor, C. R., 2002, “Methodological Approach and Reconfiguration Tool for Assembly Systems,” *CIRP Annals*, 51(1) pp. 9-13.

Ye, H., and Liang, M., 2006, “Simultaneous Modular Product Scheduling and Manufacturing Cell Reconfiguration Using a Genetic Algorithm,” *ASME Journal of Manufacturing Science and Engineering*, 128(4), pp. 984-995.