The Pennsylvania State University

The Graduate School

Industrial and Manufacturing Engineering Department

AN INVESTIGATION OF THE IMPACT OF ASSEMBLY SCHEDULING ON PRODUCT FAMILY DESIGN OUTCOMES

A Thesis in

Industrial Engineering

by

Hakan Umut Artar

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2008

The Thesis of Hakan Umut Artar was reviewed and approved* by the following:

Gül E. Kremer
Assistant Professor of Engineering Design and Industrial Engineering
Thesis Adviser

Timothy W. Simpson
Professor of Industrial Engineering and Mechanical Engineering

Richard J. Koubek
Professor of Industrial Engineering
Peter & Angela Dal Pezzo Department Head Chair
Harold & Inge Marcus Department of Industrial and Manufacturing Engineering

* Signatures are on file in the Graduate School.

ABSTRACT

Manufacturers throughout the world continuosly search for improvements in order to increase their profitability. This search becomes even more challenging when demand uncertainty and fluctuations are considered. In order to achieve better profitability, many researchers and practitioners have attempted to search for improvements in the shop floor. However, within the last two decades, it has been determined that designing product families also provides opportunities for improvements. Designing a product family that will respond to the requirements of the customer demand is essential for the survival of companies in today's competitive markets. When meeting customer requirements, companies also need to take cost initiatives into account. Overall, while many approaches have been proposed to optimize product family design for measures of cost and performance, many of these approaches fail to incorporate the complexity of manufacturing issues into family design decision-making. Assembly scheduling strategies can be considered as one of the most important of those manufacturing issues.

This research presents a computer simulation study by which the impact of assembly scheduling on the product family design outcomes is investigated. Overall, the research builds upon the foundation present in the literature, which established computer simulation as a way to investigate product family design decisions; and it extends the recent studies wherein cost and expected customer waiting times were estimated using stochastics calculations. Simulation was used in this study because it can provide a realistic model that is sensitive to demand randomness and process variations. Using the results of the simulation study, analysis of different assembly scheduling and product family design strategies can be compared for performance and profitability. It is expected that when product family design decisions are made with the consideration of assembly scheduling strategies, the outcomes at the shop floor level improve. The verification of this expectation is investigated.

It is found that alternative scheduling strategies created opportunities for managers, who are searching to improve their systems according to the customer demands. Postponement strategies were found to be useful to satisfy the customer requirements. Finding the right scheduling method for the a firm's customers could lead to higher profitability in the long run as it leads to higher customer satisfactions in terms of the two performance measures studied in this work (cost and customer response time). The make-to-stock strategy proved to produce significantly better results in terms of lead times. On the other hand, the make-to-order strategy was successful for lowering the cost of products. On the other hand, alternative product family design strategies did not offer such strong improvements for the screwdriver family studied. These results have implications for companies that seek to increase their revenue without extensively increasing their investment in the shop floor.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

First, I would like to thank Dr. Gül E. Kremer for her guidance, direction, and advice during the completion of this thesis. I would also like to thank Dr. Timothy W. Simpson for reading and reviewing the work.

Finally, I would like to acknowledge the contribution of Dr. Richard J. Koubek to my academic career.

Chapter 1

Introduction

1.1 Problem Statement

The design of product families has been investigated by several researchers in the past. While approaches have been proposed to optimize the product family design for measures of revenue or performance, most of these approaches fail to incorporate the complexity of manufacturing issues such as assembly scheduling and sequencing into product family design decision-making (Jiao et al., 2007).

In an age when consumers demand high-quality, low-priced, and customized products, the competition is not only on price but also on product variety and speed to meet the market needs. The current philosophy is to replace old products constantly with new versions, either with new and improved products or with variations of the existing products. Differentiation of products makes these new product versions possible. While being as flexible and quick to respond as possible, manufacturers have to pursue 'dynamic stability' in order to achieve the sufficient differentiation. Dynamic stability refers to process capabilities that enable responding to change in a stable and long-term yet flexible and responsive manner (Jiao et al., 2007). In these conditions, it is difficult to obtain mass production efficiency and economies of scale. This contradiction manifests itself in a new production paradigm termed as mass customization (Jiao and Tseng, 1999).

Mass customization recognizes each customer as an individual and provides each of them with 'tailor-made' product features that would be possible in the pre-industrial craft system. In the meantime, it enables customers to afford the products since modern mass production helps the product costs to remain low (Jiao and Tseng, 1999). However, the increased customization requires increases in the product variety, which will lead to increases in manufacturing complexity. Accordingly, a company aiming to stay competitive should optimize its product variety with respect to manufacturing complexity. Therefore, an effort to design and to develop product families to decrease the conflict of manufacturing complexity and the demand satisfaction with customized products is essential for success of companies in today's competitive market.

In order to meet the variety requirements in the customer demand while taking into account the manufacturing complexity, postponement (i.e., delayed product differentiation) is suggested. The postponement strategy requires the start of the production to be delayed until after the customer orders arrive. This delay is useful because it allows the differentiation of the products until later stages of the supply chain. The term postponement was coined by Zinn and Bowersox (1988). Lee and Tang (1997) argued that this delay in product differentiation helps the company to decrease the buffer inventory. Investigations of postponement are operationalizing it in two forms referred to as time postponement and form postponement (Su et al., 2005). Further details on postponement concepts are provided in the Chapter 2.

1.2 Objective

The overall objective of this thesis is to investigate time and form postponement strategies for their contribution to manufacturing floor performance metrics (lead time, cost, etc.) when various orders made up of members of a product family are considered. This investigation will help manufacturing sites to increase their efficiency and profitability. The investigation involves important issues for most manufacturing sites including design related issues, scheduling strategies and cost analysis. As performance measures, this study uses both customer response times (lead times), and cost of products and orders. The approach of using multiple performance measures is beneficial for companies making product family and assembly scheduling decisions because they can tailor their strategy for changing market conditions. In addition, this thesis differs from most studies, which are focused on optimizing the performance of individual products. A consideration of multiple products in a company product line simultaneously is more supportive for companies (Gupta and Krishnan, 1997). Overall, the investigation presented in this thesis focuses on levels of manufacturing performance metrics under alternate product designs and assembly scheduling conditions.

In order to reach these objectives, simulation models are developed to study the impact of alternative changes within the manufacturing site to quantify the improvements. Using statistical analyses of the simulation study results, the research also aims to identify the significance of alternative strategies to be implemented in the manufacturing facility.

1.3 Thesis Organization

This research builds upon the previous work in the literature related to product family design decisions and considerations of assembly scheduling. Accordingly, in Chapter 2 prior work from the literature is summarized. Concepts such as modularity, product variety, product development, manufacturing management, manufacturing cost, design for manufacturing, group technology, customization, variety management were found to be useful when product family designs were examined in the academic studies performed in the past; hence, they are included in the literature review. These concepts were defined using several definitions made by researchers in the past. In addition, they were explained using simple examples, when possible, in order to distinguish one from another. On the other hand, when the assembly scheduling was studied, the concepts, time postponement and form postponement, were encountered as the most important techniques. They are defined and explained in detail once again in Chapter 2.

In the thesis, a sample manufacturing facility was studied to see the effects of different product family design and assembly scheduling strategies. A manufacturing system that was studied in the literature provided sufficient information to adequately simulate the system. Accordingly, a simulation model was constructed and executed several times. Chapter 3 gives an overview of the manufacturing system that is used for the development of these simulation models, and explains the development of the simulation model. During the development of the simulation model, many assumptions had to be made. The rationale for each assumption is also explained in Chapter 3.

Chapter 4 provides the results of the simulation runs as well as their comparative analysis. Several alternative manufacturing conditions (scheduling strategies, demand conditions, etc.) were tested to determine the impact they had on the system. Alternative manufacturing conditions were compared to each other using the performance metrics collected by simulation runs. In addition, these results were also compared to a similar study performed by Su et al. (2005), who have also considered the product family design alternatives with the consideration of assembly scheduling using the queuing theory instead of a simulation study.

Finally, Chapter 5 summarizes this work by drawing conclusions from the findings of the study. Furthermore, this chapter also points out avenues for future research in this area.

Chapter 2

Literature Review

The argument over the ability to create product variety being within the flexibility of the manufacturing system or within the product architecture has been a major research topic during the last decade (Ulrich, 1995). For example, Hermann and Chincolkar (2000) proposed that new product design strategies should be tested by manufacturing requirements. Another suggestion by Kusiak and He (1998) was for product design strategies to be tested against production scheduling changes. Research by Jiao and Tseng (2004) targeted to maximize customer-perceived value of customization while achieving optimal design and process customizability indices. In general, these suggestions are collected under the idea of designing product variety in the shop floor. Conversely, Ulrich (1995) states that manufacturing system's ability to create variety resides not with the flexibility of the equipment in the factory, but with the architecture of the product. The research by McCutcheon et al. (1994) also suggests that companies tailor their products in order to shorten their delivery times. McCutcheon et al. (1994) referred to this as the customization-responsiveness squeeze. Salvador and Forza (2004) concluded that a trade-off exists between product variety and operational performance. Overall, these studies point to the close relationship between product variety and manufacturing system within which product variants are manufactured.

On the other hand, however, other researchers claim that product designers and process designers pursue different goals and that product designers are usually ignorant of factory capabilities. Therefore, they fail to recognize the interdependence of these two concepts (Swink, 2003). The research presented in this thesis aims to eliminate this "missing link" between designing and manufacturing for product family designs.

The literature review in this thesis is divided into two sections: (1) design, and (2) scheduling and sequencing. In the first section, the terms product family, product architecture, modularity and group technology are defined and illustrated using a simple example for clarity. The second section, scheduling and sequencing, introduces the concept known as postponement. It also focuses on why the analysis of product family design and assembly scheduling and sequencing should consider both cost benefits as well as performance benefits with alternating weights for importance.

2.1 Design

The design of a product involves establishing the functional requirements by analyzing customer needs and expectations, creating a design in order to meet these requirements, and facilitating the process to manufacture the design created (De Lit, 2003). In general, tasks undertaken to achieve a design are very complex. For example, Fung et al. (2006) characterize the incorporation of both qualitative and quantitative information regarding customer requirements and engineering characteristics as the most important problem to be solved in the new product development.

When customizing a product, two design strategies are essential. Function sharing is a design strategy that eliminates redundant properties of components by mapping components to functions. Geometric nesting is a design strategy for efficient use of space and material within the consideration of the volume of the item (Ulrich, 1995). Platform-based design is not a good fit when it comes to the balance of over-design and under-design of product variants. It is said to be negatively correlated with profitability. In order to prevent the over-design of low-end variants and under-design of high-end variants, multiple criteria such as utility and cost need to be considered at the same time. Looking at long-term benefits instead of being fixated on the fixed costs is helpful in realization of the importance of platform-based product development (Krishnan and Gupta, 2001).

Park and Simpson (2005) stated that designers need to identify and classify the production activities and resources of a production system. After the completion of the design, manufacturers establish the set of operations to realize the design into products through manufacturing. This set of operations is called process information. *Process information* can be used to improve manufacturing traceability, standardization, and quality control while decreasing the costs by utilizing the experience from previous designs and manufacturing operations. The process information requires product architecture information as an input. *Product architecture* is the arrangement of functional elements of a product into several physical building blocks, including the mapping from functional elements to physical components, and the specification of the interfaces among interacting physical components (Mikkola and Gassman, 2003). In this definition, *component* refers to any physical distinct portion of the product that embodies a core design concept and performs a well-defined function. According to Mikkola and Gassman (2003), there are two types of components: (1) standard components (2) new-to-firm components. *Standard components* are those that have been used in previous or existing architectural designs by the firm, or components that are available from the firm's library of components whereas the *new-to-firm components* refer to product-specific components that are introduced to the firm for the first time, such as modular innovations (Mikkola and Gassman, 2003).

The goals of using standard components are (Mikkola and Gassman, 2003):

1. to minimize investment as the reuse of existing components avoids significant additional investment in product development and tooling,
2. to exploit economies of scale from production volume, and
3. to preserve organizational focus leading to specialization and development capabilities.

The goals of using new-to-firm components, on the other hand, can be classified as (Mikkola and Gassman, 2003):

1. to maximize product performance with respect to holistic customer requirements,
2. to minimize the size and mass of a production – the desire for part integration in order to conserve size and mass gives rise to an integral architecture which implies that components will have to be redesigned, and

3. to minimize the variable costs of production – variables are largely determined by component mass and size. .

When the design is created using another product as a parent product, it is considered to be a member of a product family. The concept of *product family* is defined by Fan and Liu (1999) as a group of products based on a specific concept or from a standard parent product. It represents the use of common components to create versions for a product (De Lit, 2003). Product family design has been studied from several perspectives such as business strategy, marketing, manufacturing, customer needs engineering, information technology, and general management (Jiao et al. 2007). Meyer and Lehnerd (1997) state that a product family is a set of similar products derived from a common platform and possess specific features and functionality to meet specific customer requirements. Ma et al. (2002) stated that the costs and benefits of commonality in the product families should be researched and quantified for suitability. D'Souza and Simpson (2002) defined the product family as a group of related products that satisfy a variety of market niches yet share common features, components and subsystems.

The basic principles of product families are indicated as creating modular designs, avoiding variants, and standardizing at all levels of the design by De Lit et al. (2001). The *product family* is an appropriate method since modularity in product design is essentially a property of product sets offered by a company (Shirley, 1990) that typically address heterogeneous, but interrelated, customer needs (Sanchez, 1999), and that can be produced by the same process, thus allowing for economies of scope (Garud and Kumaraswamy, 1995). Sawhney (1998) stated that product family design strategies are also helpful to the companies in terms of reducing development risks and system complexities, improving the ease to upgrade products, and enhancing flexibility and responsiveness of manufacturing process. Simpson et al. (2001) defined scale-based product family design as "stretching" or "shrinking" the product platform in the directions that fits the customer requirements.

Baldwin and Clark (2000) defined modular architecture as one of the most important aspects of product family design. The product family concept is also closely related to the concept of modularity. The term *modularity* defines the act of using some guidelines and standard components among a limited number of products for another product. In the research of Mikkola and Gassman (2003), *modularity* is defined as an approach for organizing complex products and processes efficiently by decomposing complex tasks into simpler activities so they can be managed independently. *Modularity* can also include the use of encapsulated units to meet the dynamic changes being faced by their host system with the aim of identifying independent, standardized, or interchangeable units to satisfy a variety of functions (Bi and Zhang, 2001). Modularity has been argued by Pine et al. (1993) to be the key to achieve mass customization. Moreover, modularity corresponds to flexibility or changeability. The potential benefits of modularity include economies of scale, increased feasibility of product/components change, increased product variety, reduced lead time, decoupling tasks and ease of product upgrade, maintenance, repair and disposal (Coronado et al., 2004). Ulrich (1994), Feitziner and Lee (1997) and Skipworth and Harrison (2004) stated that modularity is commonly associated with assemble-to-order products where modules are inventoried and assembled upon receipt of customer order. Therefore, it can be argued that where postponement is used as a strategy, product might have high levels of modularity.

Benefits from modularity include increased feasibility of product and component change, increased product variety, reduced order lead time, decoupled risks, easier product diagnosis related to maintenance, repair and disposal (Bi and Zhang, 2001). The characteristics of modularity are the use of a finite set of components to meet infinite changes of the environment, the establishment of the module by reviewing the similarities among the components, the act of keeping as much independence of resulting work cells as possible and the use of different modules for varieties of assemblies (Bi and Zhang, 2001).

The significance of these benefits can be better recognized when the impact to the whole supply chain is examined. The research of Duray et al. (2000) pointed to the combination of four archetypes for the realization of mass customization using modularity. These archetypes are fabricators, involvers, modularizers, and assemblers. Fabricators involve the customers early in the process when unique designs can be realized or major revisions can be made in the products. Fabricators closely resemble a pure customization strategy, but they employ modularity to gain commonality of components. Involvers incorporate customer involvement in product design during the design and fabrication stages but use modularity during the assembly and delivery stages. Modularizers develop a modular approach in the design and fabrication stages, although customers do not specify their unique requirements until the assembly and use stage. Finally, assemblers provide mass customization by using modular components to present a wide range of choices to the customer (Duray et al., 2000).

The disadvantages of modularity can be redundant product architecture, excessive product similarity, and excessive capability due to designing for the most rigorous application. Forming modules early in the design process is desirable; however, it may cause modules to ignore certain constraints that appear later in the process (Kusiak, 2002). Mikkola and Gassman's (2003) research concluded that modularity management of product architectures should not be conducted in isolation of manufacturing strategy and organizational designs, especially with regards to multi-project management. The benefits of economies of substitution depend on the production volume. In cases when the demand volume is low, high levels of modularity may not be a feasible strategy. In order to explain these concepts more thoroughly, Figure 1 demonstrates how modularity works. In this figure, there are three modules with 3, 2, and 2 variants each, respectively. By taking one variant from each module, one can create new products. The possible outcomes of the combinations of these modules are given on the right side of the figure. It must be noted that interfaces allow the components to work together in the creation of the final product. The changes in the components do not prevent them from being put together in the formation of the final product.

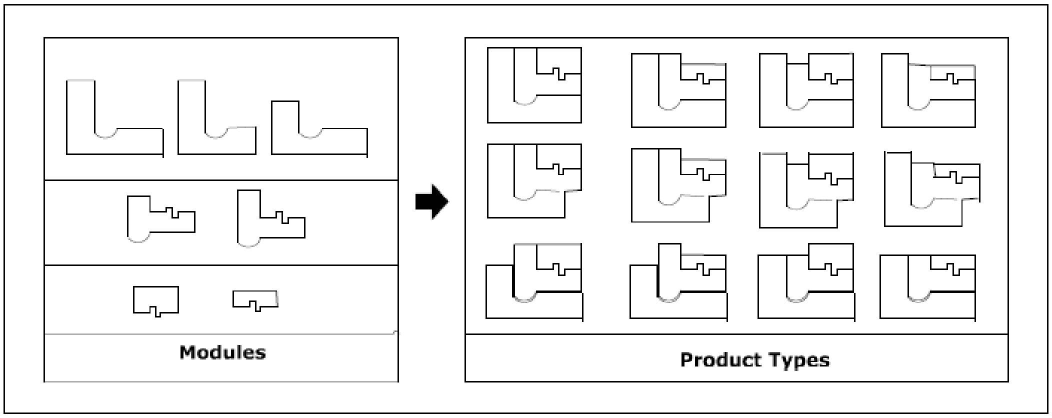


Figure 1. A Representation of Modular Architecture

The product architecture in this example is the way these three separate modules are designed to work together in order to create the final product. In other words, it is the order of the components being assembled to each other that create the final product. As an explanation of the product architecture analogy, one can think of the assembly instructions of the final product as blueprints for an architect.

The definition of the term product family can be thought as a group of products that contain multiple common components within them. The 12 possible products on the right side of the figure can be regarded as members of product family. All 12 products are variations of each other, and are the members of the same product family. The family members would differ from each other by having at least one different component. It should be noted, however, that the different members of the family would perform similar tasks given that their components are coming from the same module group, and these similar components bring the similar functionality to the system. However, there would be one capability coming from one component that makes the product member different to differentiate their functions.

The term modularity also can be explained in an easier way using the Figure 1. Modularity, in short, is the approach of decomposing complex products into simpler ones. In other words, the manufacturer of these example products produces certain components of it, knowing that they will be utilized in the production later. Modularity seeks to take advantage of the common components used by several members of this product family to create economies of scope. Therefore, the manufacturer can take advantage of stocking certain components, since they will be required for more than one product type. Consequently, the components in the example of Figure 1 are divided into three modules in order to have the option of stocking them independently.

The notions of product architecture, component, product family and modularity play an important role in this research. While studying the impact of assembly scheduling on the product family design outcomes, it is not desirable to overlook similar/related technologies that might provide viable assembly scheduling and sequencing strategies.

2.2 Scheduling and Sequencing

Potential changes executed in the scheduling strategies have the ability to affect the profitability of a manufacturing site. A generic name for all scheduling changes would be the routing information. The routing information is the way the product is produced. Examples of such information would be machines, labors and, tools fixtures, and specifications of operation sequences (Jiao and Tseng, 2004).

Overall, flexibility of the final assembly process is a key driver of the ability of a firm to offer product variety. Schierholt (2001) stated that companies are interested in configuring existing operations and manufacturing processes by exploiting similarity among product families. The consideration of design during the manufacturing process while paying attention to technical constraints has been suggested as a profit enabler (Jiao and Tseng, 2004).

One of the first major strategies in this regard is *group technology*, a method that improves manufacturing efficiency by separating components into part families (Mitrofanov, 1966). The creation of part families follows routing or design information (Burbidge, 1989; Offodile, 1991). Alizon et al. (2006) studied the use of group technology while studying the reuse of manufacturing knowledge.

Escoto et al. (1998) defined the term *group technology* as an identifier and grouper of similar parts into families, and machines into cells for advantages being in the similarity of the parts made in each cell in terms of their design and manufacturing. The benefits of group technology include material flow, cost reduction in material handling, reduction in work in progress, reduction in the cycle time and set-up time, increased manufacturing flexibility, increased quality, and increased job satisfaction (Escoto et al., 1998). The objective of grouping components was also presented as to maximize the sum of similarities between components (Catay et al., 2002). However, Catay et al. (2002) failed to mention capacity and precedence constraints and consider these constraints in their computational experience; hence, it is impossible to evaluate their results (Lourenco and Pato, 2004).

In order to increase their profit, some firms delay the final assembly until the item is ready for shipment. This implementation is called *postponement* (Ulrich, 1995). Postponement is consistently mentioned as one of the central features of mass customization (Van Hoek, 2001). It is defined by Van Hoek (2001) as an organizational strategy whereby some of the activities in the supply chain are not performed until customer orders are received. Two postponement strategies are observed in industry: (1) time postponement, and (2) form postponement (Su et al., 2005). *Time postponement (TP)* is termed for the strategy that delays the delivery until after customer orders arrive. It is also commonly referred to as a make-to-order approach (Su et al., 2005). *Form postponement (FP),* on the other hand, delays the differentiation of the products until later production stages. It requires manufacturers to stock generic (semi-finished) components from which they draw for final assembly (Su et al., 2005). The advantage of postponement is that it allows the manufacturer to restructure their process by changing the sequence of operations in their process in order to delay the point where a product is customized (Caux et al., 2006).

The work of Su et al. (2005) is an important work that evaluates product family design and assembly scheduling. This study first introduces the subject by defining TP (time postponement) and FP (form postponement) and then suggests an evaluation method of the two postponement strategies using queuing theories with M/M/1 and G/G/1 applications. In addition to the two postponement strategies, the impact of factors such as utilization rate of resources, inter-arrival time and process time variations, interest rate fluctuations, and generic component coverage in a member of a product family and the final number of products in the family are studied to identify for possible improvements. The two measuring parameters, expected customer waiting time and total cost of final products, were compared at the end of stochastic M/M/1 and G/G/1 calculations.

First, a time postponement (TP) manufacturing strategy was investigated in this work. Su et al. (2005) treated it as a system with exponential inter-arrival and service times. In this example, product types of the same product family were received at a centralized manufacturing facility. It would be processed in the facility and shipped to customers with First Come First Serve (FCFS) principle. The arrival process was independent, and the inter-arrival times come from a Poisson process with a mean arrival rate of λ. In addition, the processing was independently and identically distributed, (i.i.d.), random variables from a Poisson distribution with a mean rate of μ. With these assumptions, the system could be evaluated with an M/M/1 stochastic model, and expected customer waiting times and total cost for each product type could be obtained.

The same system was also tested as a form postponement (FP) manufacturing strategy. The production was divided into two stages. At stage 1, the generic components were produced to be held as inventory. This stage was also called make-to-stock. The expected waiting times and cost from this stage were calculated based on the analysis of Buzacott and Shanthikumar (1993). Stage 2 was designed to complete the final customizations to the products in order to exactly meet the demand. This stage was treated similarly to the TP system with single server using M/M/1. The performance parameters targeted, total cost for each product and customer waiting times, were obtained.

With the M/M/1 model results, an initial analysis of product family design was performed. In order to observe the effect of product family design, the variations in the inter-arrival times and process times were also restricted. It was observed that as the number of products in the product family increases, the total cost and the expected customer waiting time of the TP strategy remains constant. On the other hand, the total cost and the expected customer waiting time of the FP strategy increases.

Similar calculations were also performed using a G/G/1 model because a more detailed sensitivity analysis about the changes in inter-arrival time variations and process time variations were necessary. The exponential distribution in the M/M/1 queuing systems does not allow changes in the variance without changes in the mean. G/G/1 model was capable of performing that task. Total cost and expected customer waiting times were investigated once again as they were calculated in the M/M/1 model. The observations of Su et al.'s (2005) work from a G/G/1 model perspective are listed as follows.

1. Under medium to high utilization levels, higher inter-arrival time variation significantly increases the expected customer waiting time of TP but not the expected customer waiting time of FP. In other words, FP is more robust to inter-arrival time variation.
2. Higher process time variation significantly increases the expected customer waiting time of TP but not the expected customer waiting time of FP. In other words, FP is more robust to process time variation because of the ability of inventory to buffer.
3. A higher percentage of generic component coverage decreases the expected customer waiting time of FP.
4. Increasing the number of products in the product family increases both cost and expected customer waiting time of FP but not the cost and the expected customer waiting time of TP.
5. Increasing the percentage of generic component coverage and reducing the number of products in the product family in FP improves the expected customer waiting times. However, lowering

variations in process times or inter-arrival times does not significantly improve the waiting times. Therefore, the companies may get the conclusion that they should improve their products rather than their processes for better profitability.

6. Increases in interest rate increase the cost for both TP and FP. However, it has a larger impact on FP because FP has larger inventory compared to TP.

The work in this thesis is different from the work of Su et al. (2005) because it attacks the same issue with the approach of simulation. The stochastic approach used in the work of Su et al. (2005) fails to incorporate the issue of randomness to the degree that simulation can. Postponement strategies, time postponement and form postponement, are tested for better performance measurement parameters. A similar study was also performed by Van Der Zee (2004) using simulation because he also wanted to demonstrate the potential of his new strategy with different performance measures. Simulation provides a reliable analysis in these kinds of situations.

An analysis for the effectiveness of product family design and assembly scheduling strategies gives the most realistic results when several criteria are considered. The customization of products according to resources and requirements needs to be justified. This customization relies on the verification of cost-effectiveness vs. utility, cost-effectiveness vs. design changes, and cost-effectiveness vs. process variations. Customer perceived value of customization is measured by utility vs. cost that is created by multi-criteria decision making methods (Jiao and Tseng, 2004). Li and Azarm (2002) identified this issue of evaluation of candidate products using several criteria. Some of these criteria are: meeting business targets (profit and market share), meeting customer preferences, establishing a position among similar products in the market, product life cycle, and dealing with uncertainties in the market (discount rates, market size etc.).

This study analyzing the product family design and assembly scheduling strategies using a simulation study will contribute to the research conducted in the subjects mentioned in this section by combining the analysis of design and scheduling with the consideration of responsiveness (lead times) and cost. The literature review failed to find the alternative strategies incorporated in both design and scheduling issues at the same time when analyzing the impacts of these alternatives from the two performance measures' perspectives. The research of Kariyakar (2000), a computer simulation as a way to investigate product family design decisions and Su et al. (2005) wherein cost and expected customer waiting time were estimated using genetic algorithms using stochastic calculations were similar to this research in terms of their objective to improve the manufacturing system by re-evaluating product family designs and assembly scheduling methods. However, this research differentiates from the work of Kariyakar (2000) and Su et al. (2005) by the level of detail and the number of components in the product studied and the usage of computer simulation to get the two performance values.

Simpson (2004) states most problems can be treated as optimization problems as they require an analysis of large number of options. Therefore, the decisions involving the product family design and assembly scheduling should be made under the optimization of several factors such as cost and performance.

Chapter 3

Manufacturing System for Screwdriver Family

This chapter provides information about the system studied for this thesis. After the manufacturing system is described, the chapter also discusses how the simulation model representing the system was created. In addition, dependent and independent variables for this simulation study were introduced.

3.1 Description of the Manufacturing System Modeled

For the simulation model development, manufacturing system information utilized by Park (2005) is adopted. Park's dissertation had sufficient information to develop the simulation model for a screwdriver family manufacturing site.

3.1.1 Definition of the System

The model estimates that there are incoming orders with a quantity of a 100,000 screwdriver per year (Park, 2005, p. 199), and 300 working days are allowed in a year (Park, 2005, p. 199). The simulation model only runs the system for a quarter of that time, considering most companies report their financial status four times a year. Since no information was provided about the timing of the incoming orders, the model assumes that the firm receives orders with normally distributed inter-arrival times Norm (120, 30) hours. This considers the company receiving orders in a daily basis with some variation.

This screwdriver manufacturing facility produces five different types of screwdrivers. In this thesis, these products are called P1, P2, P3, P4, and P5, representing Product 1, Product 2, Product 3, Product 4, and Product 5. The bills of materials for these products are provided in Appendix A. The following sections explain some of the significant assumptions that were made during the creation of simulation models.

3.1.2 Combinations of Screwdrivers within an Order

Each order consists of different combinations of these five products. For the simulation study, the cases when more than two screwdriver types are ordered are neglected for simplicity. Thus, the firm needs to deal with orders that only include either one type of screwdriver alone or a combination of two types of screwdrivers where any combination is possible (P1 and P2, P5 and P3, etc.). The following section explains the calculation used when the number of possible order combinations are examined.

The combinations calculating the number of possible outcomes when only one product is ordered is COMBIN (5, 1) or 5 numerically. On the other hand, the combination calculating the number of possible outcomes when two product types are ordered at the same time is COMBIN (5, 2), 10. Accordingly, there are 15 possible order combinations that the firm can receive per each order arrival. In other words, an order arriving to the production facility has a 0.067 (1/15) chance of being one of the 15 possible order types.

3.1.3 Quantity of Orders

The average number of screwdrivers produced in a year was presented in Park's research; however, the distribution of this annual demand was not specified. Hence, further assumptions were required to determine the quantity of each incoming order. The simulation model assumes that the incoming orders are again normally distributed with an average of 333 screwdrivers and variance of 33 in the cases when there is only one type of screwdriver in the order. When the orders consist of two different types of screwdriver, it is assumed that the order would include two packs of different screwdrivers. Consequently, each pack would have a normally distributed number of screwdrivers with Norm (166, 16) screwdrivers.

3.1.4 Components of Screwdrivers

Once the orders arrive, the production process starts. Each screwdriver is an assembly of several components that are either products of machines such as injection, die-casting, and powder metallurgy, or products purchased from outside of the manufacturing facility. Table 1 display all the components of P1 with the detailed information such as what their machining process is, how many they are in the screwdriver, and what their costs are.

Table 1. Product 1 (P1) Bill of Materials (Park, 2005)

P1 Bill of Materials							
Subassembly	**Part Name**	**Quantity**	**Mass (gr.)**	**Material**	**Manuf. Process**	**Dimensions**	**Cost**
M1 Bit	Bit	1	13.7	M	Purchasing	L=63.5 D=6.3	$1.06
M2 Housing	Lower Housing	1	41.2	ABS	Injection	L=231	$0.94
	Upper Housing	1	42.2	ABS	Injection	L=231	
	Screw	5	0.6	M	-	L=14.9 D=2.13	$0.05
M3 Lock	Lock	1	0	P->M	Purchasing	L=40 DM=41	-
M4 Position	-	-	-	-	-	-	-
M5 Front End	-	-	-	-	-	-	-
M6 Gear	Ring Gear	1	8	P->M	Injection	L=24 D=35.5	$0.16
	Planetary Arm	1	0	M	Powder Metallur.	L=20.2 T=4	$0.58
	Planetary	3	2.9	M	Purchasing	D=11.8	$0.71
	Intermediate	1	7.6	M	Powder Metallur.	L=20.2 T=4	$0.58
	Planetary	3	0.16	P	Purchasing	D=11.8 T=4	$0.82
	Motor Coupling	1	2.2	M	Powder Metallur.	H=10.9 D=8	$0.07
M7 Shaft	Gear Train to	1	0	M	Powder Metallur.	L=40.5	$0.3
	Chuck	1	70	M	Powder Metallur.	L=55	$0.78
M8 Electric	Switch	1	12	P	Purchasing	L=36.7	$1.07
	Switch Button	1	0	P	Purchasing	L=27	-
	Battery (2.4 v)	2	22.3	-	Purchasing	L=50	$4.27
	Motor	1	64	-	Purchasing	L=51 D=27.6	$1.13
M9 Head	-	-	-	-	-	-	-

L=Length, DM=Max Diameter, Dm=Min Diameter, T=Thickness W=Width, H= Height

Table 1 presents all components that make up P1, in other words the Bill of Materials (BOM) for P1. The bill of materials for P2, P3, P4, and P5 can be found in Appendix A. The benefit of preparing the BOM is to visualize the routing information, which presents the operation precedence for fulfilling the child-parent relationship in the BOM (Jiao et al., 2005). As observed in Table 1, P1 has 17 types of different components within it. These components vary in their quantity, mass, material, and manufacturing process that they had to go through for completion before assembly or outsourcing, dimensions, and cost. Park's (2005) dissertation provides sufficient data to calculate all the process times.

According to Park (2005), there are three processes done in house in the manufacturing facility: (1) injection molding, (2) powder metallurgy, and (3) die-casting. The rest of the components are outsourced. The process times for each component depend on some specifications such as thermal diffusivity and material thickness. These calculations are done exclusively. For the injection molding process, the processing times can be calculated by the formula:

[(5+thickness^2/thermal diffusivity)/3600] (p. 199, Park, 2005) (1)

The setup time takes 2 hours and requires high skilled labor, which costs $90/hour. All this information is available in Park's (2005) dissertation. For powder metallurgy, the dissertation does not give enough information to calculate the processing times. However, some information was provided in the BOM of the five types of screwdrivers; hence, the information found in the BOMs was used. The setup time takes 1 hour. The powder metallurgy setup requires high skilled labor, which costs $90/hour. For the die-casting operations, the processing time is 1/4300 hour for simple components, 1/3000 hour for medium components, and 1/700 hour for complex components. The setup time is 1 hour for simple, 1.5 hours for medium, and 2 hours for complex. Die-casting requires high skilled labor, which costs $90/hour. In addition to the components coming from the three types of workstations, there are also components that are outsourced. The processing times for purchasing components were determined to be 1 second assuming that the outsourced components would arrive instantaneously.

3.1.5 Capacity of the System

Park (2005) did not have any information on the number of machines in the manufacturing facility. However, it provided information about the yearly demand. So, it was necessary to balance the system finding a reasonable number of machines for each workstation once the simulation model was created. Initially, the simulation model assumed to have only one machine for each type of workstation. It was determined that additional machines were necessary in the injection and die-casting processes in order to have the system function in a way to meet the demand within reasonable wait times for customers. The number of machines in system's workstations was alternated to reach a utilization level that would be able to produce the annual demand in a balanced way. At the end of this balancing, there were still bottlenecks in the system. The production was being completed in most cases but there were opportunities to improve the performance. Nonetheless, the manufacturing site could afford to have them considering the yearly demand being manufactured. The final numbers of machines at the end of the balancing trials were 8 injection molding, 10 die-casting, and 1 powder metallurgy machines.

3.1.6 Assembly

After the parts are processed by the machines in the model, they go to the final stage of their manufacturing: assembly. Each product type must collect all its components in order to be assembled. When they are missing one or more components, they are waiting in a queue for their missing components. Once all components are done with their processes, they are ready to be assembled. The components, ready to be assembled after their machining process, are queued regardless of their product types. The placement of the components in appropriate queue is done with respect to FIFO law according to their product type. The assembly process uses a constant amount of time regardless of which product or how many components are assembled. In Park's dissertation, this constant time is provided as 114 sec (Park, 2005, p. [199]). In order to add variability to the assembly time, the normal distribution with a mean value of 114 seconds and a standard deviation 2 seconds was used to represent the assemble delay.

3.2 Simulating the System

The main purpose of simulating this system is to find the lead times (customer response or waiting times) and the total cost of each product and order. The simulation tool, ARENA version 9.0, is selected to produce this simulation model because of its convenience in terms of programming and its ability for object oriented modeling. This section explains the reasoning behind the choices made while the simulation model was being created. The model creation is explained in two major sections. Section 3.2.1 explains the general flow in the original model that uses the make-to-order strategy for scheduling the assembly of parts into finished products, and orders. Section 3.2.2 explains how the system handles the inventory.

3.2.1 Make-to-order Model

It is decided that the model would start with the creation of entities representing the incoming orders to the manufacturing system. Each order consists of different combinations of 5 products, P1-P5. The model assigns one of the 15 possible combinations randomly and with equal chance. ARENA achieves this by assigning one of the 15 alternative combinations randomly and then creating 15 different routes of manufacturing paths for each of the alternatives. Once the order combination is assigned and placed in the path that was predefined by the model, the entity is cloned to represent the screwdrivers to be manufactured rather than the orders. For example, in the case when the order consists of two types of screwdriver, two groups of entities representing the two different types of screwdrivers are created to meet the number of products in the order. Consequently, entities start to represent actual screwdrivers waiting to be produced, and they all go to a pool where they are filtered according to their product types being one of the five possibilities. This filter serves as their process information identifier. The routing information of the products is visible at the point of filtration. After order combination is determined, the model once again clones entities per all the screwdrivers in order, this time, representing the components. The number of clones varies according to the number of components of each of the 5 product types from 17 to 42 since P1 has 17 and P5 has 42 components within it. This allows the model to send the components to different machines. Components are processed by only one machine before they are assembled.

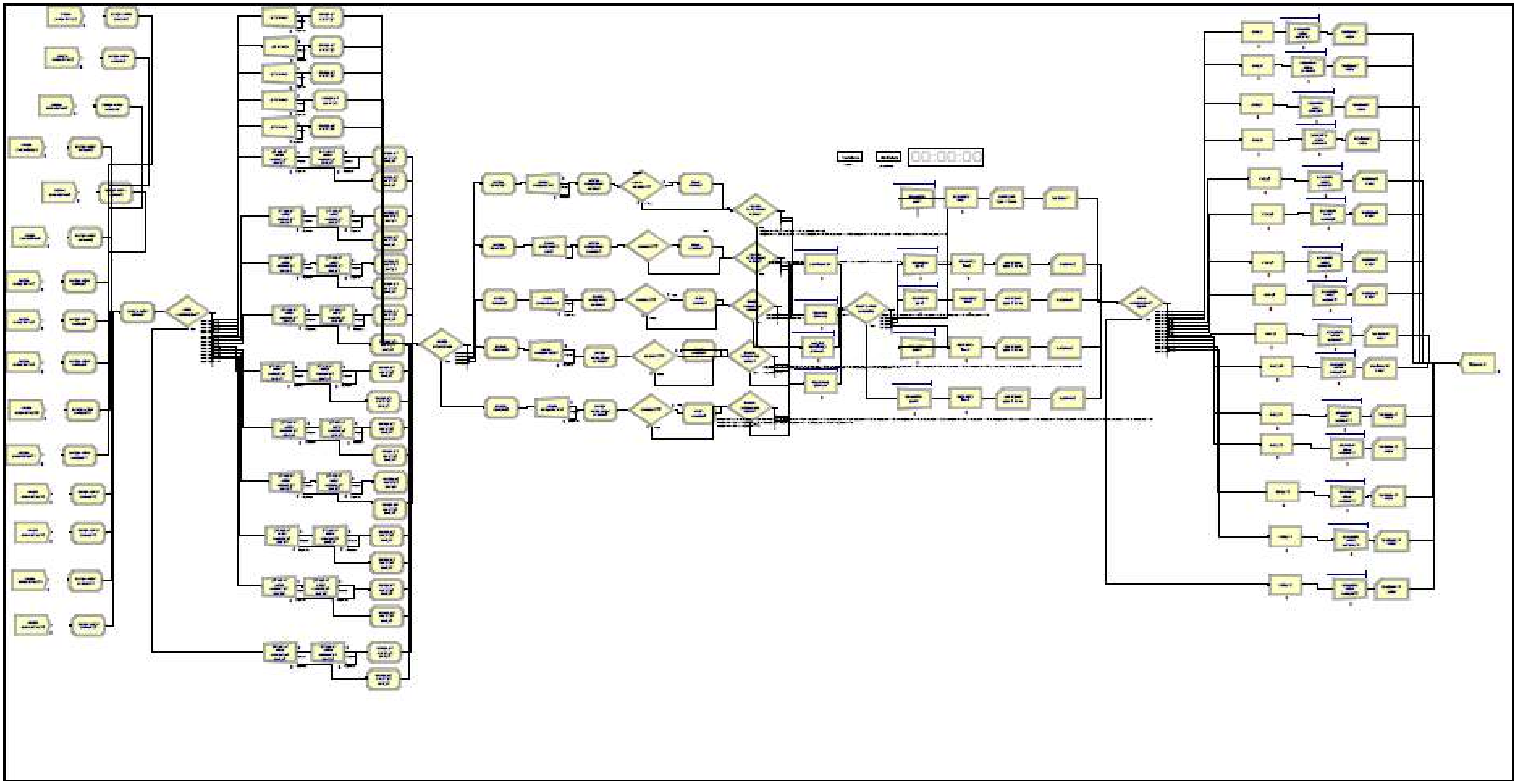
At this stage of the modeling, it is useful to distinguish the screwdrivers so that at later stages there is no confusion of components being assembled to wrong screwdrivers. The simulation model also needs to make sure that the components created are assembled into proper product types. The model stops cloning components when all the components of that specific product type are created. It questions whether the right number of components is already produced. If this is the case, then the model resets the components to zero for the next incoming entity (screwdriver). After all the components are created, they are once again filtered according to the machines that are going to process them. At this stage, the components are placed in queues in front of the machines waiting for their turn. Using attributes for all different components, the model keeps track of the processing times for each specific component. After the machines process components, the parts go through another filter where they are divided according to their part type into a queue. This queue holds the components while waiting for other components before they are assembled.

After the machines process the components, the parts go through another filter where they are divided according to their part type into a queue. This queue holds the components waiting for other components before they are assembled. Then, the model picks the components that belong to the same screwdriver and puts them together in the amount of time that was pre-specified. Before all components of a screwdriver are present, the process time for assembling process does not start. Once a screwdriver is completed, it proceeds to the area where it is waiting for other screwdrivers that are also a part of the same order as it. When all screwdrivers of the same order arrive, the order is ready to be shipped out of the system. In the simulation methodology, this process is called disposal of entities.

The assembled screwdriver entities are recorded with their processed entity numbers and their lead times before they are disposed from the model. Accordingly, although entities are cloned to become components in early stages of the model, this does not affect the final statistics since the components are batched together in an assembly line prior to recording statistics. For recording the lead times, the statistics that are necessary are the average lead times for each order to be processed after they enter to the system as orders. The model collects this data recording the moment of inclusion of the entity representing the order at the moment of order reception to the system through its disposal at the moment of the completion of production. Finally, after all the statistics are recorded, the entities are ready to be disposed by the model from five different points – one for each of the five product types.

Figure 2 is the model built using the data from Park (2005). This system has a make-to-order strategy and does not keep inventory for future orders. From right to left, the reception of orders and the separation of these orders into 15 possible order combinations and then into five members of the product family can be observed in this figure. After the production of these five members of the product family, observed in the middle section of the figure, the model collects the products once again into 15 order combinations to get them ready for shipment. Siman Codes for this model is presented in Appendix B.

Figure 2. The Make-to-order Model



3.2.2 Make-to-stock Model

As in the case of the make-to-order model, the order handling in the make-to-stock model requires the system to let orders be the entities in the model. Later, orders become the actual screwdrivers, and go though processes. However, the make-to-stock model is different from the make-to-order system in the way it handles the common components of the product family. Eight of components are selected as common components after studying the product BOMs, and they are produced at all times independently from the order arrivals.

The make-to-stock model handles the common components without processing them in the machines because they are already processed and stocked in the inventory. The model follows the same general principles. However, the model yields different lead times and costs because of the way common components are handled. The inventory system was determined using properties of (s, S) policy. Section 4.1.1.2 gives further details on this (s,S) policy.

3.3 Experimental Design

After the completion of the simulation model according to the make-to-order strategy, alternative product family design and assembly scheduling strategies needed to be investigated. Therefore, first, additional strategies that were previously studied by researchers were searched from the published literature. According to the findings of this study, the experiment was designed. The variables of the investigation are grouped into two classes: independent variables and dependent variables.

3.3.1 Independent Variables

The simulation model was run several times in order to check the potential effects of the changes that could be applied to the system. The main concern during the evaluation of the effects of these changes was how they would modify the total cost of the production and the average lead time of the product family. These changes can be classified under four main categories, and are related to (1) the scheduling strategies, (2) the resources, (3) the design of the product family, and (4) the demand as follows.

3.3.1.1 The Scheduling Strategies

Scheduling strategies such as time postponement (TP) and form postponement (FP) can be tested to observe how the system reacts in terms of total production cost and average lead time. The time postponement strategy requires orders to be put into processing as soon as they arrive. Therefore, this strategy would enter the orders in the manufacturing system immediately as they arrive. On the other hand, the form postponement strategy works on the common components even during the times when they are not to be used immediately. When an order is received some of the components of those orders are already processed, and only assembly is required. This scheduling strategy is also called make-to-stock. The simulation model can be used to analyze these two strategies and their impacts on the performance measures.

3.3.1.2 Resources

Park (2005) provided little information about the capacity of the manufacturing site; therefore, it was necessary to bring the manufacturing site to the level of capacity it would take to meet the demand when the simulation model was designed. However, considering the impact of number of resources is a decision manufacturing facilities face frequently. Simulation is a useful tool when evaluating such decisions. Accordingly, when searching for a more efficient system especially in terms of total costs and lead times, it would be beneficial to test the system for alternative numbers of resources. The simulation model estimates the total cost and average lead time outcomes for the product family incorporating different numbers of machines in the system.

3.3.1.3 Design of the Product Family

The design of the product family can be alternated in order to improve the manufacturing facility performance in terms of total cost and average lead time. The screwdriver case can illustrate a good example. In this case, the manufacturing site aims to meet the demand for five different members in the screwdriver product family. It is also known that these screwdrivers are only different from each other in terms of their capabilities. The five products of the family are numbered from 1 to 5, 1 being the least capable and 5 being the most. Technically, a customer who requests screwdriver P1 would not have any problem receiving screwdriver P2 provided that this change does not affect the price. Therefore, a manufacturing site should recursively investigate the design of the family according to the changing conditions of the market. It might be more profitable for the manufacturing facility to have fewer than five product types in the family for improving their performance measures (e.g., total cost and average lead time). Although they might be increasing the cost, by offering more functionality to the customers, they might still increase their profits thanks to the advantages of creating a more efficient shop floor. The simulation study can provide information about whether this is true or not.

3.3.1.4 Demand

Given the fluctuating market conditions, the variation of the demand is something that manufacturing facilities should pay attention to in order to maintain their long-term profitability. The simulation model created to screen the production of the screwdriver product family can also provide information that examines how the variability in the demand affects the manufacturing facility. Alternating demand quantities were tested to observe the influences on total cost and average lead time. In addition, some experiments might be useful in order to measure the performance of the manufacturing system in terms of total cost and average lead time when the orders received only contain one member of the screwdriver product family. The simulation model created is based on the assumptions that these orders contain one or at most two types of screwdrivers. It might be profitable for the manufacturing facility to modify their order reception policy accordingly so that they run a more efficient plant with orders involving one type of screwdriver at a time.

Table 2 briefly lists the changes that were simulated and executed on the model in order to observe their effects when searching for the manufacturing condition with high performance results. Listed changes were performed independently.

Table 2. Independent Variables

Change Category	Independent Variables	Alternative
Time Postponement vs. Form Postponement	Orders with interarrival time Norm (120,30) hr, 1 order at a time Orders assigned to all 15 types of orders Process according to make-to-order strategy	Orders with interarrival time Norm (120,30) hr, 1 order at a time Orders assigned to all 15 types of orders Process according to make-to-stock strategy
Resources (# of machines)	Injection : 8 machines Powder Metallurgy : 1 machine Die-Casting : 10 machines	Injection : 9 machines Powder Metallurgy : 1 machine Die-Casting : 10 machines
	Injection : 8 machines Powder Metallurgy : 1 machine Die-Casting : 10 machines	Injection : 8 machines Powder Metallurgy : 1 machine Die-Casting : 11 machines
Product Family Design	5 types of screwdrivers (1-2-3-4-5) in the screwdriver product family	4 types of screwdrivers (2-3-4-5) in the screwdriver product family Type 1 orders to be replaced by Type 2 orders
	5 types of screwdrivers (1-2-3-4-5) In the screwdriver product family	3 types of screwdrivers (3-4-5) In the screwdriver product family Type 1 and 2 orders to be replaced by Type 3 orders
Demand	Yearly demand of 100,000 Screwdrivers Yearly demand of 100,000 Screwdrivers Orders with interarrival time Norm (120,30) hr.	Yearly demand of 120,000 Screwdrivers Yearly demand of 80,000 Screwdrivers Orders with interarrival time Norm (120,40) hr.
Restricted Demand	Orders are taken in the form of 1 or 2 screwdriver types at a time	Orders are taken in the form of only 1 screwdriver type at a time

3.3.2 Dependent Variables

The work by Su et al. (2005) focused on the two parameters when comparing the different alternative strategies: (1) the average lead time, and (2) the total production cost for different members of the product family. In addition to these two performance measures, work-in-process inventory, utilization ratios, and average set-up time can also be used in different situations. Another study was performed also using simulation on the same subject. The work of Nagarur and Azeem (1999) considered three criteria: (1) makespan, (2) average machine utilization, and (3) factory productivity as criteria of performance. Swaminathan and Tayur (1998) studied the impact of changes on variance of product demands, capacity, and number of products in the product family solely in terms of cost. Table 3 summarizes some other performance metrics used in recent computer simulation studies of manufacturing system investigations.

Table 3. Performance Measures Used in Related Previous Research

References	Performance Measure
Bruccoleri et al. (2003)	Average Flow Time
Lambert et al. (2006)	Average lead time, average work in process, inventory, average set-up time
Salum (2000)	Total manufacturing lead time
Saitou (2002)	Production cost
Kattan et al. (2003)	Average cycle time, utilization ratios
Golany et al. (1999)	Work-in-process
Heikkila and Koivo (1998)	Process time
Li et al. (1996)	Flow-time
Rhee (1998)	Mean response time

Park (2005), on the other hand, was focused on cost analysis of running a facility that produces screwdrivers. He focused on the product family issue from an expense-revenue perspective; and accordingly, he did not incorporate the time relevant performance metrics. In this thesis, both cost and time measures are considered, and the following measures are adopted: (1) the total production cost, and (2) the average lead time for all five screwdriver types.

3.3.2.1 Total Production Cost

The total production cost includes the cost of each component, the machine processing costs, and the assembly cost along with labor costs. It is important to measure the cost performance of a manufacturing system because the manufacturing facility only continues to exist as long as it meets the demand for a reasonable cost. Therefore, in order to observe the influence of the experimental changes on the manufacturing performance, the total production cost was used.

3.3.2.2 Average Lead Time

In addition to total production cost, the average lead time was used to see how the system reacted to the changes modeled using computer simulation. The average lead time is the total time screwdrivers spend in the manufacturing facility. It includes the time that components spend in the queues before the machining and assembly and the process time on the machines and the assembly time. Using different scheduling strategies, the average lead time was examined. The simulation model was constructed to investigate the assembly scheduling and product family design combinations that will yield lowest average lead time while keeping the total production cost within a range that is acceptable to the customers.

Chapter 4

Results

4.1 Introduction

Several simulation runs were performed after the model was constructed with alternative assembly scheduling and product family design strategies. Each run was planned to last 500 hours, which represents one financial quarter, equivalent to a company's three months of production time. The 500 hours duration was selected accounting for the steady state conditions and considering that it would be long enough for the system to reflect the reaction of the system to the changes studied and it would also be short enough for the Arena version 9.0 to handle it despite the hardware and run time restrictions.

Random orders were processed in the facility under varying conditions. First, a full-factorial experimental design was adopted to investigate the most important three independent variables: demand level, postponement strategy, and the product family design. Then, for several independent variables, the single model modification mentioned in Table 2 was studied. At the end of the runs, the statistics of the performance measures collected in the simulation were compared to the original system's statistics collected in the model that is created according to the data from Park (2005) in order to verify responses of the system to the changes investigated. The results obtained are tabulated. A table comparing the results of the simulation runs with different postponement strategies, make-to-order vs. make-to-stock, under several demand-arrivals, resource availability, and product family design conditions are created and added to this study. The lead time (the amount of time the screwdrivers spend in the system from the moment of order arrival to the moment of order shipment) performances are presented along with the total cost of screwdrivers. The calculation of the cost measure is further explained in Section 4.1.1.

In order to recognize the impacts of these different strategies, the data collected from the Arena version 9.0 output files were compared for each order type. Since the incoming orders were the same and equal numbers of observations for each order type were represented in the system, these statistical significance tests were performed (t-tests). In addition to the t-tests, confidence intervals were showed, which were calculated by using standard deviations and average values. Minitab version 15 was used for this task. The results of these comparisons can be found in Appendix C.

4.1.1 Calculation of Cost

The statistics collected by the simulation model were used in order to predict the costs of orders and products. Section 4.1.1.1 and Section 4.1.1.2 provide a detailed explanation of these calculations.

4.1.1.1 Total Cost of Screwdrivers for Make-to-order

For the make-to-order assembly scheduling model, the total costs (TC) of screwdrivers were calculated by adding the total component cost (TCC) to the WIP inventory cost (WIPIC).

$$TC = TCC + WIPIC \quad (2)$$

The total component cost of a screwdriver was available in the BOM appropriate. Screwdrivers would not be complete without all components necessary to form them. Therefore, TCC is treated as a fixed cost. The costs of components included what their purchasing price is when they are known to be purchased by the screwdriver manufacturer. In addition, they also included their processing cost when they are known to be produced by the manufacturer using the Injection, Die-Casting, and Powder Metallurgy workstations. The processing costs for these workstations were calculated according to the dimensions of the components. Since these calculations were already done when the BOMs were prepared, this thesis used the calculation results when the costs for components were needed.

In addition to the fixed component costs, there is also cost associated with the inventory held in the manufacturing facility during the assembly of screwdrivers from the moment they are requested by the customers to the moment of shipment. Therefore, the statistics of how much time the components spent in the system were collected in the simulation model. This amount of time was multiplied by the 20 percent of the total component cost. The assumption of 20 percent was chosen to reflect a reasonable percentage that represents the cost of having components in the manufacturing site throughout the production stage. Inventory holding cost calculations required to incorporate this cost to the total cost of the products.

4.1.1.2 Total Cost of Screwdrivers for Make-to-stock

For the make-to-stock assembly scheduling model, the total costs (TC) of screwdrivers were calculated by adding the total component cost (TCC) to the WIP inventory cost (WIPIC) and the average inventory holding cost (AIHC).

$$TC = TCC + WIPIC + AIHC \qquad (3)$$

The fixed component cost, once again, was taken from the BOM information. Differently from the make-to-order approach, for the make-to-stock approach, the total costs of uncommon components were utilized. The make-to-stock model's assumption of receiving the cost of a common component from inventory when it is required for a screwdriver is used to rationalize the calculation of fixed costs. The inclusion of the WIPIC into the total cost calculation was done in a similar manner as was done for make-to-order approach. The statistics for the time duration starting from the order arrival until its shipment were recorded. The 20 percent inventory holding rate assumption was used one more time to calculate the cost of that WIP to the manufacturing company. The total costs of uncommon components were again calculated similar to the make-to-order approach for the WIPIC calculation. The time the uncommon components spend in the system from the moment of order arrival to the moment of order shipment and the total cost of the uncommon components of the screwdrivers in question were used considering the inventory holding rate assumption.

In addition to the TCC and the WIPIC, the make-to-stock approach required a third element to realistically reflect the situation. This element is referred as the average inventory holding cost in this study. For the calculation of AIHC, the average inventory held by the company at any time is investigated. An (s, S) policy that guarantees the inventory to be able to meet the demand at all types of order arrivals was determined as a base for this calculation. It was assumed that the company should be able to pull the components of all 5 types of screwdrivers for each incoming orders. Therefore, the total cost of common components was multiplied by 5 in order to get the average inventory holding cost.

4.1.2 Full Factorial Design

The three independent variables from Table 2 demand level, postponement strategies, and different product family design alternatives, were selected to include in a full factorial experimental design to fully investigate (with two-way and three-way interactions) how the alternative strategies would impact the system. The impacts of these variables would be the most important ones for the selected performance measures, lead times and cost. The results of the simulation models allowing the completion of this full factorial study can be found in the following sections and Appendix D.

The demand level was selected to be in this full factorial design because it was important to investigate how the system would handle fluctuations, within the range of the 20% of the original demand. Companies are responsible to keep their production line stable in such situations. In addition to the fluctuated demand levels, different postponement strategies were also studied. Previous research studied in the literature review suggested that postponement strategies have potential to improve the manufacturing systems. Using the simulation model created for this research, this study could analyze this improvement. Finally, alternative product family design strategies, in other words, placing different number of products in the product family, was included in the full factorial study. The screwdrivers studied allow alternative product family options. Therefore, studying the alternative designs using the simulation system would verify the benefits of selecting the most suitable product family design alternative.

The full factorial design study allows this work to determine the significance of the impact of the changes studied. The statistics tool MANOVA (multivariate analysis of variance) is used to study the significance. MANOVA is used because there are two dependent variables that are of interest, and it was important to investigate the impact of alternative strategies on the dependent variables simultaneously. In general, usage of MANOVA reduces the likelihood of Type I error in multiple dependent variable conditions in comparison to a serious of ANOVA. In order to use MANOVA, the average values of the fifteen order types for lead time and cost values was investigated for the selected independent variables. Table 4 provides a summary of these values that the simulation model produced for each model representing different demand levels, different postponement strategies, and different product family design alternatives in the system.

Table 4. Full Factorial Design Experimentation Results

Demand Level	Postponement Strategy	Number of Screwdrivers in the Family	Lead Times (hrs.)	Order Costs ($)
80 K Demand	make-to-order	3 Members	7.17	42.13
80 K Demand	make-to-order	4 Members	6.89	40.78
80 K Demand	make-to-order	5 Members	6.75	39.73
80 K Demand	make-to-stock	3 Members	2.56	90.09
80 K Demand	make-to-stock	4 Members	2.46	89.70
80 K Demand	make-to-stock	5 Members	2.38	89.25
100 K Demand	make-to-order	3 Members	11.86	58.77
100 K Demand	make-to-order	4 Members	11.33	56.47
100 K Demand	make-to-order	5 Members	11.13	54.99
100 K Demand	make-to-stock	3 Members	2.80	90.77
100 K Demand	make-to-stock	4 Members	2.94	91.06
100 K Demand	make-to-stock	5 Members	2.88	90.61
120 K Demand	make-to-order	3 Members	11.97	58.36
120 K Demand	make-to-order	4 Members	10.94	40.78
120 K Demand	make-to-order	5 Members	18.09	77.29
120 K Demand	make-to-stock	3 Members	3.63	93.02
120 K Demand	make-to-stock	4 Members	3.76	93.28
120 K Demand	make-to-stock	5 Members	3.60	92.61

Table 5 presents the results of MANOVA analysis. The significance levels are represented with p-values in this table. It is observed that when the two dependent variables, lead times and costs, are considered, the independent variable, demand level, produced p-values that are smaller than 0.05. This means when different demand levels are studied using the simulation model, the model produces results that are significantly different from each other.

Table 5 also includes results of the significance tests of the other two independent variables, different postponement strategies and different product family designs. For the postponement strategies, MANOVA found a strong significance between the two strategies studied. P-value of 0.00 is the indication of this strong significance. Finally, different product family design alternatives were studied using MANOVA. P-values found for this independent variable were higher than 0.05. Such high values represent no significance between different product family design strategies.

Table 5. MANOVA Results Table

General Linear Model: Lead times and Costs vs. Demand Level, Postponement Strategies, Product Family Design Alternatives

MANOVA for Demand Level

s = 2 m = -0.5 n = 4.5

Test Criterion	DF Statistic	F	Num	Denom	P
Wilks'	0.39634	3.236	4	22	0.031
Lawley-Hotelling	1.40534	3.513	4	20	0.025
Pillai's	0.65034	2.891	4	24	0.044
Roy's	1.31584				

MANOVA for Postponement Strategies

s = 1 m = 0.0 n = 4.5

Test Criterion	DF Statistic	F	Num	Denom	P
Wilks'	0.00551	993.569	2	11	0.000
Lawley-Hotelling	180.64893	993.569	2	11	0.000
Pillai's	0.99449	993.569	2	11	0.000
Roy's	180.64893				

MANOVA for Product Family Design Alternatives

s = 2 m = -0.5 n = 4.5

Test Criterion	DF Statistic	F	Num	Denom	P
Wilks'	0.80266	0.639	4	22	0.640
Lawley-Hotelling	0.23522	0.588	4	20	0.675
Pillai's	0.20588	0.689	4	24	0.607
Roy's	0.17411				

In addition to these MANOVA results, some tests using ANOVA were also used to check how the independent variables impact the dependent variables individually. MANOVA provided analysis of dependent variables (lead times and costs) together. However, it was also important to perform analysis to uncover the impact of experimental conditions on each dependent variable separately. This is due to the fact that some companies might consider only one of the dependent variables considered here as significant for their success (e.g., using only total cost for a cost leadership strategy). Appendix E lists these tables. According to the ANOVA results, when lead times are considered, the number of products in the family does not cause significant difference. On the other hand, the two other independent variables demand level and postponement strategies cause significant differences in the system. The ANOVA table considering the cost as the dependent variable concluded the same result.

Furthermore, two-way interaction effects of independent variables were studied. ANOVA for both dependent variables, lead times and costs, found insignificant interaction impact. This means when the different combinations of two independent variables are applied to the system, it is observed that no significant impact to the dependent variables is observed.

Su et al. (2005) concluded that higher percentage of generic component coverage leads to decreases in the lead times in the make-to-order strategy. However, the MANOVA table results shows that the number of product in the product families, in other words, different percentage levels of generic component coverage did not cause significant impacts on the system. From this perspective, the simulation study presented in the thesis differed from Su et al.'s (2005) paper. However, when the two remaining independent variables were investigated and compared with the findings of the algorithm used by Su et al. (2005), it was found that the two studies concluded similarly in that the demand levels and postponement strategies both had significant effects on dependent variables in this study as they were mentioned to have had in Su et al.'s (2005) paper.

In Tables 6 and 7, values for the total lead times and total costs of all orders and products calculated by the simulation model are presented. The data used in the ANOVA and MANOVA analyses were taken directly from the results tables such as Tables 6 and 7. As seen in these tables, lead time and cost values are provided for all order types (made up of different product combinations) as well as for each product type (P1-P5). Section 4.2.1 approaches to the same tables from a different perspective and comment on these findings.

Table 6. The Lead Time Values for the Original 100 K Demand Model with the Make-to-order and Make-to-stock Strategies

100K Demand make-to-order Order Lead Times	**Hr.**	**100K Demand make-to-stock Order Lead Times**	**Hr.**
Order Type 1 (p1)	8.3475	Order Type 1 (p1)	3.1693
Order Type 2 (p2)	4.3808	Order Type 2 (p2)	2.7829
Order Type 3 (p3)	5.3732	Order Type 3 (p3)	3.1147
Order Type 4 (p4)	14.457	Order Type 4 (p4)	2.2009
Order Type 5 (p5)	18.145	Order Type 5 (p5)	2.5595
Order Type 6 (p1-p2)	8.2266	Order Type 6 (p1-p2)	3.4245
Order Type 7 (p1-p3)	5.8881	Order Type 7 (p1-p3)	3.1782
Order Type 8 (p1-p4)	11.846	Order Type 8 (p1-p4)	2.4092
Order Type 9 (p1-p5)	11.178	Order Type 9 (p1-p5)	3.0915
Order Type 10 (p2-p3)	7.4973	Order Type 10 (p2-p3)	2.9293
Order Type 11 (p2-p4)	16.213	Order Type 11 (p2-p4)	4.1524
Order Type 12 (p2-p5)	16.374	Order Type 12 (p2-p5)	2.7828
Order Type 13 (p3-p4)	16.079	Order Type 13 (p3-p4)	2.5489
Order Type 14 (p3-p5)	7.1395	Order Type 14 (p3-p5)	2.3381
Order Type 15 (p4-p5)	15.825	Order Type 15 (p4-p5)	2.4583
100K Demand make-to-order Product Lead Times		**100K Demand make-to-stock Product Lead Times**	
P1	5.2074	P1	1.3272
P2	5.1616	P2	1.6403
P3	4.8355	P3	1.7495
P4	10.823	P4	1.8469
P5	10.946	P5	2.0893

*values are in hours.

Table 7. The Cost Values for the original 100 K Demand Model with the Make-to-order and Make-to-stock Strategies

100 K Demand make-to-order Order Cost	**$**	**$**	**$**	**100K Demand make-to-stock Order Cost**	**$**	**$**	**$**	**$**
	Fixed Cost	**WIP Inventory Cost**	**Total Cost**		**Fixed Cost**	**WIP Inventory Cost**	**Average Inventory Holding Cost**	**Total Cost**
Order Type 1 (p1)	12.8	21.3	34.1	Order Type 1 (p1)	11.42	7.24	69.15	87.81
Order Type 2 (p2)	14.2	12.4	26.6	Order Type 2 (p2)	12.82	7.14	69.15	89.11
Order Type 3 (p3)	14.8	15.9	30.6	Order Type 3 (p3)	13.44	8.37	69.15	90.96
Order Type 4 (p4)	18.8	54.5	73.3	Order Type 4 (p4)	14.08	6.20	69.15	89.43
Order Type 5 (p5)	20.8	75.6	96.5	Order Type 5 (p5)	15.78	8.08	69.15	93.01
Order Type 6 (p1-p2)	13.5	22.2	35.6	Order Type 6 (p1-p2)	12.12	8.30	69.15	89.57
Order Type 7 (p1-p3)	13.8	16.2	30.0	Order Type 7 (p1-p3)	12.43	7.90	69.15	89.48
Order Type 8 (p1-p4)	15.8	37.4	53.3	Order Type 8 (p1-p4)	12.75	6.14	69.15	88.04
Order Type 9 (p1-p5)	20.8	46.6	67.4	Order Type 9 (p1-p5)	15.78	9.76	69.15	94.69
Order Type 10 (p2-p3)	14.5	21.7	36.1	Order Type 10 (p2-p3)	13.13	7.69	69.15	89.97
Order Type 11 (p2-p4)	16.5	53.5	70.0	Order Type 11 (p2-p4)	13.45	11.17	69.15	93.77
Order Type 12 (p2-p5)	17.5	57.3	74.8	Order Type 12 (p2-p5)	14.3	7.96	69.15	91.41
Order Type 13 (p3-p4)	16.8	54.0	70.8	Order Type 13 (p3-p4)	13.76	7.01	69.15	89.92
Order Type 14 (p3-p5)	17.8	25.4	43.2	Order Type 14 (p3-p5)	14.61	6.83	69.15	90.59
Order Type 15 (p4-p5)	19.8	62.8	82.6	Order Type 15 (p4-p5)	14.93	7.34	69.15	91.42
100 K Demand make-to-order Product Cost				**100K Demand make-to-stock Product Cost**				
P1	12.77	13.3	26.1	P1	11.42	3.03	69.15	83.60
P2	14.17	14.6	28.8	P2	12.82	4.21	69.15	86.18
P3	14.75	14.3	29.0	P3	13.44	4.70	69.15	87.29
P4	18.84	40.8	59.6	P4	14.08	5.20	69.15	88.43
P5	20.84	45.6	66.5	P5	15.78	6.59	69.15	91.52

*values are in US dollars.

The rest of the tables carrying the statistics of the remaining models can be found in Section 4.2. In addition to these tables, Section, 4.2 provides a discussion on the results of the model to the different conditions executed. In addition, the results of the statistics collected in these tables are also compared to the results obtained in the Su et al.'s (2005) study.

4.2 Interpretation of Results

After analyzing the results from simulation models in ANOVA and MANOVA tables, comparisons of different strategies were selected and performed. The main reason of this type of comparison was seeing the effect of independent variables on all order and product types of the system rather than seeing their effect on the system in general through averaged lead time and cost values. The products in the system had a large variation in the number of components they carried (i.e., 17 – 42). This large variation required more detailed analysis than provided in the ANOVA and MANOVA tests.

The results of simulation runs are analyzed in the following sections according to the changes performed to the original screwdriver manufacturing system.

4.2.1 Make-to-stock Strategy vs. Make-to-order Strategy of the 100 K Demand Model

Table 6 is helpful when comparing the lead times of different order types produced using alternative strategies. This table indicates that overall the make-to-stock strategy gives lower lead times. The level of significance was measured using t-tests. Table 8 is a sample t-test analysis that is created using Minitab software. The p-values being equal to 0.00 prove the significance of the difference between the two (make-to-order vs. make-to-stock) strategies of the 100 K Demand model.

Table 8. Sample t-Test Results for the Make-to-stock Strategy vs. Make-to-order Strategy of the 100 K Demand Model: Order Lead Times

100 K Demand Model t-test				
Make-to-order vs. make-to-stock:	**ORDER LEAD TIMES**			
	N	**Mean**	**StDev**	**SE Mean**
make-to-order Lead Times	15	11.13	4.72	1.22
make-to-stock Lead Times	15	2.88	0.51	0.13
Difference	15	8.26	4.83	1.25
95% CI for mean difference: (5.58, 10.93)				
t-Test of mean difference = 0 (vs not = 0): t-Value = 6.63 p-Value = 0.000				

In the make-to-stock strategy, the common components are produced regularly without being affected by the order arrivals. When the orders arrive, the common components produced earlier get attached to the uncommon components of the ordered products. The production of those uncommon components does not start until the order has arrived. Since the amount of production after the order arrival is decreased, this decrease also reflects to the order lead times. Table 6 lists lead time statistics that are received from the two simulation models investigating different scheduling (make-to-order vs. make-to-stock) approaches for orders and products.

In addition to the lead times, the total production cost of orders and products were also used as performance measures for the system under investigation. Table 7 summarizes the cost calculations using the same models with the two competing postponement strategies: TP and FP. With respect to cost, the make-to-order strategy was advantageous. The costs for all possible order combinations with make-to-order strategy were lower than the costs for all possible order combinations with make-to-stock strategy. These comparisons were verified with the t-tests performed. The p-value of 0.00 was obtained. It can be argued that having the common components available in the inventory significantly impacted the production costs of the system. Same analysis can be performed from a cost of product perspective rather than cost of orders. Again, the costs for all possible product types with the make-to-order strategy were smaller than the costs for all possible product types with the make-to-stock strategy. These statistical values were verified using the confidence interval data created while conducting t-tests.

This analysis can also be compared to the findings of Su et al. (2005) study. In this study, the total cost of the time postponement strategy, in other words, the make-to-order strategy is investigated with performance measures of total product cost and total customer waiting time (total lead time in this study). Su et al. (2005) had concluded that a higher percentage of generic component coverage decreases the expected customer waiting time of FP. Therefore, this simulation study found similar results from the lead time perspective.

4.2.2 Alternate Demand Scenarios

In order to test the flexibility of the system to respond to a variation in the demand level, the number of products in the incoming orders was modified to be 20 percent more than it currently is (120 K Demand Model). Simulation results were helpful to see how this change would affect the system. For the lead times of both orders and products, the make-to-stock strategy produced better results. The make-to-stock strategy produced lead times that are lower than the make-to-order strategy. This observation was very helpful because, when the 100 K Demand Model lead times were compared to 120 K Demand Model lead times; a better understanding of the benefits of the make-to-stock strategy can be attained. It is observed that in the higher demand market, represented as the 120 K Demand Model, the differences between the make-to-order strategy and the make-to-stock strategy lead times get smaller. This suggests that the make-to-stock strategy creates a more suitable environment for production expansion occasions. Table 9 displays the lead time values of the 120 K Model with make-to-stock and make-to-order strategies.

Table 9. The Lead Time Values of the 120 K Model with Make-to-stock and Make-to-order Strategies

120K Demand make-to-order Order Lead Times	**Hr.**	**120K Demand make-to-stock Order Lead Times**	**Hr.**
Order Type 1 (p1)	19.071	Order Type 1 (p1)	3.6602
Order Type 2 (p2)	15.660	Order Type 2 (p2)	3.8896
Order Type 3 (p3)	10.403	Order Type 3 (p3)	3.8057
Order Type 4 (p4)	16.746	Order Type 4 (p4)	2.3216
Order Type 5 (p5)	19.793	Order Type 5 (p5)	3.2769
Order Type 6 (p1-p2)	10.814	Order Type 6 (p1-p2)	4.3296
Order Type 7 (p1-p3)	20.450	Order Type 7 (p1-p3)	3.2678
Order Type 8 (p1-p4)	15.747	Order Type 8 (p1-p4)	3.1766
Order Type 9 (p1-p5)	19.923	Order Type 9 (p1-p5)	4.5882
Order Type 10 (p2-p3)	14.685	Order Type 10 (p2-p3)	3.9337
Order Type 11 (p2-p4)	28.585	Order Type 11 (p2-p4)	4.3661
Order Type 12 (p2-p5)	17.867	Order Type 12 (p2-p5)	2.5484
Order Type 13 (p3-p4)	22.465	Order Type 13 (p3-p4)	3.9778
Order Type 14 (p3-p5)	16.031	Order Type 14 (p3-p5)	3.6558
Order Type 15 (p4-p5)	23.096	Order Type 15 (p4-p5)	3.1712
120K Demand make-to-order Product Lead Times		**120K Demand make-to-stock Product Lead Times**	
P1	11.715	P1	1.6694
P2	11.437	P2	2.019
P3	11.039	P3	2.2521
P4	15.751	P4	2.2475
P5	14.084	P5	2.5625

When the costs of orders and products in the 120 K Demand Model were investigated, it is found that, cost-wise, the make-to-stock option was not performing for the favor of the company. The costs with the make-to-stock strategy are much higher simply because of all the inventory holding costs of components waiting in the stock. Therefore, although the system is saving a significant amount of time to respond the customer, it suffers from high inventory holding costs. However, when the make-to-stock strategy order and product costs for 100 K Demand and 120 K Demand are compared, it was observed that make-to-stock strategy is more suitable for the increased demand level. Larger order amounts and larger order variations make the inventory holding more valuable because it eases the production providing the common components from inventory shelves. Hence, these larger order amounts and larger order variations justify the use of the make-to-stock strategy more than the 100 K Demand. Table 10 displays the cost values for the 120 K Demand model with the make-to-order and make-to-stock strategies. The statistical analysis regarding the comparisons of 120 K Demand with the make-to-order approach and the make-to-stock approach and the original model with regular demand and the make-to-order approach and the make-to-stock approach can be found along with t-test results in Appendix C.

Table 10. The Cost Values for the 120 K Demand Model with the Make-to-order and Make-to-stock Strategies

120 K Demand make-to-order Order Cost	$	$	$	120K Demand make-to-stock Order Cost	$	$	$	$
	Fixed Cost	WIP Inventory Cost	Total Cost		Fixed Cost	WIP Inventory Cost	Average Inventory Holding Cost	Total Cost
Order Type 1 (p1)	12.8	48.82	61.62	Order Type 1 (p1)	11.42	8.36	69.15	88.93
Order Type 2 (p2)	14.2	44.47	58.67	Order Type 2 (p2)	12.82	9.97	69.15	91.94
Order Type 3 (p3)	14.8	30.79	45.59	Order Type 3 (p3)	13.44	10.23	69.15	92.82
Order Type 4 (p4)	18.8	62.96	81.76	Order Type 4 (p4)	14.08	6.54	69.15	89.77
Order Type 5 (p5)	20.8	82.34	103.14	Order Type 5 (p5)	15.78	10.34	69.15	95.27
Order Type 6 (p1-p2)	13.5	29.20	42.70	Order Type 6 (p1-p2)	12.12	10.49	69.15	91.76
Order Type 7 (p1-p3)	13.8	56.44	70.24	Order Type 7 (p1-p3)	12.43	8.12	69.15	89.70
Order Type 8 (p1-p4)	15.8	49.76	65.56	Order Type 8 (p1-p4)	12.75	8.10	69.15	90.00
Order Type 9 (p1-p5)	20.8	82.88	103.68	Order Type 9 (p1-p5)	15.78	14.48	69.15	99.41
Order Type 10 (p2-p3)	14.5	42.59	57.09	Order Type 10 (p2-p3)	13.13	10.33	69.15	92.61
Order Type 11 (p2-p4)	16.5	94.33	110.83	Order Type 11 (p2-p4)	13.45	11.74	69.15	94.34
Order Type 12 (p2-p5)	17.5	62.53	80.03	Order Type 12 (p2-p5)	14.3	7.29	69.15	90.74
Order Type 13 (p3-p4)	16.8	75.48	92.28	Order Type 13 (p3-p4)	13.76	10.95	69.15	93.86
Order Type 14 (p3-p5)	17.8	57.07	74.87	Order Type 14 (p3-p5)	14.61	10.68	69.15	94.44
Order Type 15 (p4-p5)	19.8	91.46	111.26	Order Type 15 (p4-p5)	14.93	9.47	69.15	93.55
120 K Demand make-to-order Product Cost				**120K Demand make-to-stock Product Cost**				
P1	12.77	29.92	42.69	P1	11.42	3.81	69.15	84.38
P2	14.17	32.41	46.58	P2	12.82	5.18	69.15	87.15
P3	14.75	32.57	47.32	P3	13.44	6.05	69.15	88.64
P4	18.84	59.35	78.19	P4	14.08	6.33	69.15	89.56
P5	20.84	58.70	79.54	P5	15.78	8.09	69.15	93.02

Another simulation model was created to test the decrease of incoming orders by 20 percent down to 80 K Demand Model level. Tables 11 and 12 present the related comparative analysis. Order and product lead times were again compared for make-to-order and make-to-stock strategies. The make-to-stock strategy produced the screwdrivers in a significantly shorter lead time that the make-to-order strategy. The t-test for the significance produced a p-value of 0.00. In addition to this analysis, a comparison of the two models, 100 K Demand Model and 80 K Demand Model was also performed in order to see the effect of the change on the product and order lead times. It was found that 80 K Model created lower lead time statistics. Given that the model achieves the lower demand by decreasing the number of screwdrivers in the orders, this was straightforward. See Appendix B for further details of the comparison statistics.

However, the make-to-stock strategy was not as successful with respect to cost. The costs were considerably higher when they were compared to the cost results of the make-to-order strategy. When the make-to-stock strategy costs for the 80 K Demand Model and the 100 K Demand Model are compared, it is revealed that the 80 K Demand Model does not perform as well as 100 K Demand Model. Again, since the inventory holding cost is already present in the model, the company would be better off producing as much as they can to decrease the average inventory cost per order.

Table 11. The Lead Time Values for the 80 K Demand Model with the Make-to-order and Make-to-stock Strategies

80K Demand make-to-order Order Lead Times	**Hr.**	**80K Demand make-to-stock Order Lead Times**	**Hr.**
Order Type 1 (p1)	5.356	Order Type 1 (p1)	2.4382
Order Type 2 (p2)	3.217	Order Type 2 (p2)	3.0099
Order Type 3 (p3)	4.3564	Order Type 3 (p3)	2.4277
Order Type 4 (p4)	9.303	Order Type 4 (p4)	1.605
Order Type 5 (p5)	10.567	Order Type 5 (p5)	2.0967
Order Type 6 (p1-p2)	5.5647	Order Type 6 (p1-p2)	3.3047
Order Type 7 (p1-p3)	3.8109	Order Type 7 (p1-p3)	2.2901
Order Type 8 (p1-p4)	8.6307	Order Type 8 (p1-p4)	1.8703
Order Type 9 (p1-p5)	6.638	Order Type 9 (p1-p5)	2.3922
Order Type 10 (p2-p3)	4.1191	Order Type 10 (p2-p3)	3.1035
Order Type 11 (p2-p4)	9.3873	Order Type 11 (p2-p4)	3.2448
Order Type 12 (p2-p5)	7.3778	Order Type 12 (p2-p5)	1.7963
Order Type 13 (p3-p4)	6.3718	Order Type 13 (p3-p4)	2.1325
Order Type 14 (p3-p5)	7.4556	Order Type 14 (p3-p5)	2.4864
Order Type 15 (p4-p5)	9.0679	Order Type 15 (p4-p5)	1.5283
80K Demand make-to-order Product Lead Times		**80K Demand make-to-stock Product Lead Times**	
P1	3.0955	P1	1.1122
P2	2.8619	P2	1.7394
P3	2.6531	P3	1.4686
P4	5.5402	P4	1.3831
P5	5.6733	P5	1.6038

Table 12. The Cost Values for the 80 K Demand Model with the Make-to-order and Make-to-stock Strategies

80 K Demand make-to-order Order Cost	**$**	**$**	**$**	**80K Demand make-to-stock Order Cost**	**$**	**$**	**$**	**$**
	Fixed Cost	**WIP Inventory Cost**	**Total Cost**		**Fixed Cost**	**WIP Inventory Cost**	**Average Inventory Holding Cost**	**Total Cost**
Order Type 1 (p1)	12.8	13.71	26.51	Order Type 1 (p1)	11.42	5.57	69.15	86.14
Order Type 2 (p2)	14.2	9.14	23.34	Order Type 2 (p2)	12.82	7.72	69.15	89.69
Order Type 3 (p3)	14.8	12.89	27.69	Order Type 3 (p3)	13.44	6.53	69.15	89.12
Order Type 4 (p4)	18.8	34.98	53.78	Order Type 4 (p4)	14.08	4.52	69.15	87.75
Order Type 5 (p5)	20.8	43.96	64.76	Order Type 5 (p5)	15.78	6.62	69.15	91.55
Order Type 6 (p1-p2)	13.5	15.02	28.52	Order Type 6 (p1-p2)	12.12	8.01	69.15	89.28
Order Type 7 (p1-p3)	13.8	10.52	24.32	Order Type 7 (p1-p3)	12.43	5.69	69.15	87.27
Order Type 8 (p1-p4)	15.8	27.27	43.07	Order Type 8 (p1-p4)	12.75	4.77	69.15	86.67
Order Type 9 (p1-p5)	20.8	27.61	48.41	Order Type 9 (p1-p5)	15.78	7.55	69.15	92.48
Order Type 10 (p2-p3)	14.5	11.95	26.45	Order Type 10 (p2-p3)	13.13	8.15	69.15	90.43
Order Type 11 (p2-p4)	16.5	30.98	47.48	Order Type 11 (p2-p4)	13.45	8.73	69.15	91.33
Order Type 12 (p2-p5)	17.5	25.82	43.32	Order Type 12 (p2-p5)	14.3	5.14	69.15	88.59
Order Type 13 (p3-p4)	16.8	21.41	38.21	Order Type 13 (p3-p4)	13.76	5.87	69.15	88.78
Order Type 14 (p3-p5)	17.8	26.54	44.34	Order Type 14 (p3-p5)	14.61	7.27	69.15	91.03
Order Type 15 (p4-p5)	19.8	35.91	55.71	Order Type 15 (p4-p5)	14.93	4.56	69.15	88.64
80 K Demand make-to-order Product Cost				**80K Demand make-to-stock Product Cost**				
P1	12.77	7.91	20.68	P1	11.42	2.54	69.15	83.110265
P2	14.17	8.11	22.28	P2	12.82	4.46	69.15	86.429822
P3	14.75	7.83	22.58	P3	13.44	3.95	69.15	86.537597
P4	18.84	20.88	39.72	P4	14.08	3.89	69.15	87.12481
P5	20.84	23.65	44.49	P5	15.78	5.06	69.15	89.991593

When this system with alternative demand scenario was compared to the findings of Su et al.'s (2005) findings, it was found that that the make-to-stock strategy is more robust against the alternative scenarios in terms of the costs and lead times of the production. However, the make-to-order strategy in the Su et al.'s (2005) work is more responsive to the changes in demand. Su et al. (2005) explains this deviation in costs by the involvement of the WIP inventory and final product inventory. However, the make-to-stock simulation model in this thesis takes advantage of the inventory and decreases the impact of the alternated demand on the system.

4.2.3 Alternate Capacity Strategies

After the experimental conditions for investigating how the system would react to increases and decreases in the incoming demands were investigated, the different system capacity alternatives was studied. The system had a number of resources that constituted the system capacity. How the system would perform when these resource levels altered needed to be questioned in order to have a real sense of the facility's capability. Different number of resources was considered. Two models were created to investigate these independent variables. The thesis names these models Extra Injection Model and Extra Die-Casting Model.

The Extra Injection Model had 9 injection machines. When the Extra Injection Model was run, it was found that the make-to-stock strategy once again performed better in terms of order and product lead times than the make-to-order strategy. The lead times of the make-to-stock strategy were lower than the ones of the make-to-order strategy. It was observed that between the two values, there was a 95% CI for mean differences (3.525, 7.608) (see Appendix C). When the Extra Injection Model statistics were compared to 100 K Demand Model, it is seen that the additional injection machine was helpful in decreasing order and product lead times. The statistics of the Extra Injection Models can be found in Table 13.

On the other hand, the make-to-stock strategy with the additional injection machine was not as advantageous in terms of cost. This strategy was producing the product in a more costly way. It is important to keep in mind that there was an inventory cost that created a high percentage of this cost difference.

When the Extra Injection model results were compared to the original 100 K Demand Model results with the make-to-stock strategy order and product costs, it is detected that the model with the additional resource was able to decrease the costs under the make-to-order strategy. Considering that additional machines would decrease the WIP inventory and cost related to it, it is straightforward to justify this improvement. However, with the make-to-stock strategy, the Extra Injection model created more costly screwdrivers than the 100 K Demand model. The difference between the two model's cost statistics was within the 10% of the total costs. Remembering that the make-to-stock strategy's costs are mainly affected by the inventory holding, this comparison can be explained by faster inventory production of the Extra Injection model. In addition to creating the screwdrivers in a more quick fashion, this model also created WIP inventory in a faster way. This leads to a higher average inventory holding cost for the manufacturing company and influences the costs. These statistics can be found in the Tables 13 and 14 Extra Injection Model Lead Times and Extra Injection Model Total Costs tables. The comparisons of these statistics can be seen in Appendix C.

Table 13. The Lead Time Values for the Extra Injection Model with the Make-to-order and Make-to-stock Strategies

Extra Injection make-to-order Order Lead Times	Hr.	Extra Injection make-to-stock Order Lead Times	Hr.
Order Type 1 (p1)	4.4968	Order Type 1 (p1)	3.1693
Order Type 2 (p2)	5.166	Order Type 2 (p2)	2.7829
Order Type 3 (p3)	4.279	Order Type 3 (p3)	3.1147
Order Type 4 (p4)	11.954	Order Type 4 (p4)	2.2009
Order Type 5 (p5)	14.436	Order Type 5 (p5)	2.5595
Order Type 6 (p1-p2)	4.6975	Order Type 6 (p1-p2)	3.4245
Order Type 7 (p1-p3)	4.327	Order Type 7 (p1-p3)	3.1782
Order Type 8 (p1-p4)	7.2289	Order Type 8 (p1-p4)	2.4092
Order Type 9 (p1-p5)	11.218	Order Type 9 (p1-p5)	3.0915
Order Type 10 (p2-p3)	5.8539	Order Type 10 (p2-p3)	2.9293
Order Type 11 (p2-p4)	11.004	Order Type 11 (p2-p4)	4.1524
Order Type 12 (p2-p5)	9.8674	Order Type 12 (p2-p5)	2.7828
Order Type 13 (p3-p4)	9.2216	Order Type 13 (p3-p4)	2.5489
Order Type 14 (p3-p5)	9.9816	Order Type 14 (p3-p5)	2.3381
Order Type 15 (p4-p5)	12.913	Order Type 15 (p4-p5)	2.4583
Extra Injection make-to-order Product Lead Times		**Extra Injection make-to-stock Product Lead Times**	
P1	2.2802	P1	1.3272
P2	3.4566	P2	1.6403
P3	2.9008	P3	1.7495
P4	6.4936	P4	1.8469
P5	8.5087	P5	2.0893

Table 14. The Cost Values for the Extra Injection Model with the Make-to-order and Make-to-stock Strategies

Extra Injection make-to-order Order Cost	**$**	**$**	**$**	**Extra Injection make-to-stock Order Cost**	**$**	**$**	**$**	**$**
	Fixed Cost	**WIP Inventory Cost**	**Total Cost**		**Fixed Cost**	**WIP Inventory Cost**	**Average Inventory Holding Cost**	**Total Cost**
Order Type 1 (p1)	12.8	11.51	24.31	Order Type 1 (p1)	11.42	7.24	69.15	87.81
Order Type 2 (p2)	14.2	14.67	28.87	Order Type 2 (p2)	12.82	7.14	69.15	89.11
Order Type 3 (p3)	14.8	12.67	27.47	Order Type 3 (p3)	13.44	8.37	69.15	90.96
Order Type 4 (p4)	18.8	44.95	63.75	Order Type 4 (p4)	14.08	6.20	69.15	89.43
Order Type 5 (p5)	20.8	60.05	80.85	Order Type 5 (p5)	15.78	8.08	69.15	93.01
Order Type 6 (p1-p2)	13.5	12.68	26.18	Order Type 6 (p1-p2)	12.12	8.30	69.15	89.57
Order Type 7 (p1-p3)	13.8	11.94	25.74	Order Type 7 (p1-p3)	12.43	7.90	69.15	89.48
Order Type 8 (p1-p4)	15.8	22.84	38.64	Order Type 8 (p1-p4)	12.75	6.14	69.15	88.04
Order Type 9 (p1-p5)	20.8	46.67	67.47	Order Type 9 (p1-p5)	15.78	9.76	69.15	94.69
Order Type 10 (p2-p3)	14.5	16.98	31.48	Order Type 10 (p2-p3)	13.13	7.69	69.15	89.97
Order Type 11 (p2-p4)	16.5	36.31	52.81	Order Type 11 (p2-p4)	13.45	11.17	69.15	93.77
Order Type 12 (p2-p5)	17.5	34.54	52.04	Order Type 12 (p2-p5)	14.3	7.96	69.15	91.41
Order Type 13 (p3-p4)	16.8	30.98	47.78	Order Type 13 (p3-p4)	13.76	7.01	69.15	89.92
Order Type 14 (p3-p5)	17.8	35.53	53.33	Order Type 14 (p3-p5)	14.61	6.83	69.15	90.59
Order Type 15 (p4-p5)	19.8	51.14	70.94	Order Type 15 (p4-p5)	14.93	7.34	69.15	91.42
Extra Injection make-to-order Product Cost				**Extra Injection make-to-stock Product Cost**				
P1	12.77	5.82	18.59	P1	11.42	3.03	69.15	83.60
P2	14.17	9.80	23.97	P2	12.82	4.21	69.15	86.18
P3	14.75	8.56	23.31	P3	13.44	4.70	69.15	87.29
P4	18.84	24.47	43.31	P4	14.08	5.20	69.15	88.43
P5	20.84	35.46	56.30	P5	15.78	6.59	69.15	91.52

The powder metallurgy operation was not studied further because the data related to this resource did not necessitate further study because the utilization levels did not reach very high values. The other model that dealt with the resources is the Extra Die-Casting model. In the 100 K Demand model, there is 10 die-casting machines and this amount is sufficient to meet the incoming demand. The Extra Die-Casting Model tests for an additional die-casting model. It needs to be noted that the die-casting process was only required by the two most complicated screwdrivers, P4 and P5. The make-to-stock strategy with the additional die-casting machine performed better than the make-to-stock strategy without the additional die-casting machine in terms of lead times. On the other hand, the make-to-order strategy with the additional die-casting machines provided lower costs that the make-to-stock strategy with the additional die-casting machine. However, when the comparisons of the Extra Die-Casting Models and original 100 K Demand Models were performed, it was found that the lead times and costs were not affected. Product types 4 and especially 5, however, gave clues that if there were more products that required components from the die-casting processes, then the extra die-casting machine could provide improved lead times but increased costs. Tables 15 and 16 lay out the results of this simulation runs of the Extra Die-Casting Models. The insignificance is caused by the fact that die-casting operations are mostly performed before order arrivals for the inventory stocking purpose (see Appendix C for t-tests verifying the significance of the results' comparisons).

Table 15. The Lead Time Values for the Extra Die-Casting Model with the Make-to-order and Make-to-stock Strategies

Extra Die-Casting make-to-order Order Lead Times	**Hr.**	**Extra Die-Casting make-to-stock Order Lead Times**	**Hr.**
Order Type 1 (p1)	4.6613	Order Type 1 (p1)	3.1693
Order Type 2 (p2)	5.2053	Order Type 2 (p2)	2.7829
Order Type 3 (p3)	6.3289	Order Type 3 (p3)	3.1147
Order Type 4 (p4)	14.358	Order Type 4 (p4)	2.2009
Order Type 5 (p5)	14.827	Order Type 5 (p5)	2.5595
Order Type 6 (p1-p2)	6.8588	Order Type 6 (p1-p2)	3.4245
Order Type 7 (p1-p3)	5.4232	Order Type 7 (p1-p3)	3.1782
Order Type 8 (p1-p4)	9.8123	Order Type 8 (p1-p4)	2.4092
Order Type 9 (p1-p5)	10.551	Order Type 9 (p1-p5)	3.0915
Order Type 10 (p2-p3)	5.0952	Order Type 10 (p2-p3)	2.9293
Order Type 11 (p2-p4)	10.644	Order Type 11 (p2-p4)	4.1524
Order Type 12 (p2-p5)	8.0792	Order Type 12 (p2-p5)	2.7828
Order Type 13 (p3-p4)	11.483	Order Type 13 (p3-p4)	2.5489
Order Type 14 (p3-p5)	10.631	Order Type 14 (p3-p5)	2.3381
Order Type 15 (p4-p5)	12.886	Order Type 15 (p4-p5)	2.4583
Extra Die-Casting make-to-order Product Lead Times		**Extra Die-Casting make-to-stock Product Lead Times**	
P1	2.6214	P1	1.3272
P2	3.4668	P2	1.6403
P3	4.6419	P3	1.7495
P4	8.941	P4	1.8469
P5	8.4587	P5	2.0893

Table 16. The Cost Values for the Extra Die-Casting Model with the Make-to-order and Make-to-stock Strategies

Extra Die-Casting make-to-order Order Cost	$	$	$	Extra Die-Casting make-to-stock Order Cost	$	$	$	$
	Fixed Cost	WIP Inventory Cost	Total Cost		Fixed Cost	WIP Inventory Cost	Average Inventory Holding Cost	Total Cost
Order Type 1 (p1)	12.8	11.93	24.73	Order Type 1 (p1)	11.42	7.24	69.15	87.81
Order Type 2 (p2)	14.2	14.78	28.98	Order Type 2 (p2)	12.82	7.14	69.15	89.11
Order Type 3 (p3)	14.8	18.73	33.53	Order Type 3 (p3)	13.44	8.37	69.15	90.96
Order Type 4 (p4)	18.8	53.99	72.79	Order Type 4 (p4)	14.08	6.20	69.15	89.43
Order Type 5 (p5)	20.8	61.68	82.48	Order Type 5 (p5)	15.78	8.08	69.15	93.01
Order Type 6 (p1-p2)	13.5	18.52	32.02	Order Type 6 (p1-p2)	12.12	8.30	69.15	89.57
Order Type 7 (p1-p3)	13.8	14.97	28.77	Order Type 7 (p1-p3)	12.43	7.90	69.15	89.48
Order Type 8 (p1-p4)	15.8	31.01	46.81	Order Type 8 (p1-p4)	12.75	6.14	69.15	88.04
Order Type 9 (p1-p5)	20.8	43.89	64.69	Order Type 9 (p1-p5)	15.78	9.76	69.15	94.69
Order Type 10 (p2-p3)	14.5	14.78	29.28	Order Type 10 (p2-p3)	13.13	7.69	69.15	89.97
Order Type 11 (p2-p4)	16.5	35.13	51.63	Order Type 11 (p2-p4)	13.45	11.17	69.15	93.77
Order Type 12 (p2-p5)	17.5	28.28	45.78	Order Type 12 (p2-p5)	14.3	7.96	69.15	91.41
Order Type 13 (p3-p4)	16.8	38.58	55.38	Order Type 13 (p3-p4)	13.76	7.01	69.15	89.92
Order Type 14 (p3-p5)	17.8	37.85	55.65	Order Type 14 (p3-p5)	14.61	6.83	69.15	90.59
Order Type 15 (p4-p5)	19.8	51.03	70.83	Order Type 15 (p4-p5)	14.93	7.34	69.15	91.42
Extra Die-Casting make-to-order Product Cost				**Extra Die-Casting make-to-stock Product Cost**				
P1	12.77	6.70	19.47	P1	11.42	3.03	69.15	83.60
P2	14.17	9.82	23.99	P2	12.82	4.21	69.15	86.18
P3	14.75	13.69	28.44	P3	13.44	4.70	69.15	87.29
P4	18.84	33.69	52.53	P4	14.08	5.20	69.15	88.43
P5	20.84	35.26	56.10	P5	15.78	6.59	69.15	91.52

Su et al. (2005) studied different capacities by considering the utilization levels of the system. They concluded that with uncertain demand, a high utilization level increases the lead times of the TP, the make-to-order strategy. Therefore, additional machines, reduced utilization levels, help companies to reduce the lead times. On the other hand, Su et al. (2005) stated that in the FP, the make-to-stock strategy, high utilizations did not affect the lead times. It needs to be noted that the research by Su et al. (2005) consider a system with higher percentage of common components compared to the one used in this thesis.

4.2.4 Alternate Demand Variability Scenarios

After the possible outcomes of additional machines were studied, variability in the incoming demand was investigated. Su et al. (2005) studied this factor to a limited degree. This thesis goes into more detail. The thesis refers to this model as Increased Demand Variation Model. For this model, the distribution of the incoming orders was modified to be Norm (120, 40) hr. as opposed to Norm (120, 30) hr. in the 100 K Demand model. When the simulation models with TP and FP, in other words, make-to-order and make-to-stock strategies, were compared, it was found that the make-to-stock strategy has outdone the make-to-order strategy in the product and order lead times thanks to common components that are produced prior the reception of orders. This was not surprising because it was in line with the simulation runs that are previously performed. The statistics collected from the Increased Demand Variation model with make-to-order and make-to-stock strategies are displayed in Table 17.

Table 17. The Lead Time Values for the Increased Demand Variation Model

with the Make-to-order and Make-to-stock Strategies

Increased Demand Variation make-to-order Order Lead Times	**Hr.**	**Increased Demand Variation make-to-stock Order Lead Times**	**Hr.**
Order Type 1 (p1)	9.1424	Order Type 1 (p1)	2.9269
Order Type 2 (p2)	7.8412	Order Type 2 (p2)	2.9533
Order Type 3 (p3)	5.6319	Order Type 3 (p3)	3.2644
Order Type 4 (p4)	11.858	Order Type 4 (p4)	2.8434
Order Type 5 (p5)	16.182	Order Type 5 (p5)	2.5635
Order Type 6 (p1-p2)	5.8842	Order Type 6 (p1-p2)	3.3923
Order Type 7 (p1-p3)	9.1103	Order Type 7 (p1-p3)	2.9636
Order Type 8 (p1-p4)	9.6679	Order Type 8 (p1-p4)	2.5021
Order Type 9 (p1-p5)	10.376	Order Type 9 (p1-p5)	3.0482
Order Type 10 (p2-p3)	7.0872	Order Type 10 (p2-p3)	2.8438
Order Type 11 (p2-p4)	12.637	Order Type 11 (p2-p4)	4.1689
Order Type 12 (p2-p5)	9.3738	Order Type 12 (p2-p5)	2.7529
Order Type 13 (p3-p4)	11.657	Order Type 13 (p3-p4)	2.9516
Order Type 14 (p3-p5)	11.183	Order Type 14 (p3-p5)	2.3714
Order Type 15 (p4-p5)	11.136	Order Type 15 (p4-p5)	2.1079
Increased Demand Variation make-to-order Product Lead Times		**Increased Demand Variation** make-to-stock Product Lead Times	
P1	5.7034	P1	1.2417
P2	4.8477	P2	1.7462
P3	5.5091	P3	1.7266
P4	7.392	P4	2.1013
P5	8.9028	P5	2.0343

Moreover, when the Increased Demand Variation Model order and product lead times were compared to the original 100 K Demand Model order and product costs, it is observed that the 100 K Demand Model had shorter wait times (lead times). This was reasonable because the increased variability in orders would affect the WIP times and cause delays in the production. The variability caused the lead times to increase. However, the make-to-stock approach was able to lessen the effects of it. Statistics related to these comparisons can be found in Table 17 (see Appendix C for t-tests).

Table 18. The Cost Values for the Increased Demand Variation Model with the Make-to-order and Make-to-stock Strategies

Increased Demand Variation make-to-order Order Cost	**$**	**$**	**$**	**Increased Demand Variation make-to-stock Order Cost**	**$**	**$**	**$**	**$**
	Fixed Cost	**WIP Inventory Cost**	**Total Cost**		**Fixed Cost**	**WIP Inventory Cost**	**Average Inventory Holding Cost**	**Total Cost**
Order Type 1 (p1)	12.8	23.40	36.20	Order Type 1 (p1)	11.42	6.69	69.15	87.26
Order Type 2 (p2)	14.2	22.27	36.47	Order Type 2 (p2)	12.82	7.57	69.15	89.54
Order Type 3 (p3)	14.8	16.67	31.47	Order Type 3 (p3)	13.44	8.77	69.15	91.36
Order Type 4 (p4)	18.8	44.59	63.39	Order Type 4 (p4)	14.08	8.01	69.15	91.24
Order Type 5 (p5)	20.8	67.32	88.12	Order Type 5 (p5)	15.78	8.09	69.15	93.02
Order Type 6 (p1-p2)	13.5	15.89	29.39	Order Type 6 (p1-p2)	12.12	8.22	69.15	89.49
Order Type 7 (p1-p3)	13.8	25.14	38.94	Order Type 7 (p1-p3)	12.43	7.37	69.15	88.95
Order Type 8 (p1-p4)	15.8	30.55	46.35	Order Type 8 (p1-p4)	12.75	6.38	69.15	88.28
Order Type 9 (p1-p5)	20.8	43.16	63.96	Order Type 9 (p1-p5)	15.78	9.62	69.15	94.55
Order Type 10 (p2-p3)	14.5	20.55	35.05	Order Type 10 (p2-p3)	13.13	7.47	69.15	89.75
Order Type 11 (p2-p4)	16.5	41.70	58.20	Order Type 11 (p2-p4)	13.45	11.21	69.15	93.81
Order Type 12 (p2-p5)	17.5	32.81	50.31	Order Type 12 (p2-p5)	14.3	7.87	69.15	91.32
Order Type 13 (p3-p4)	16.8	39.17	55.97	Order Type 13 (p3-p4)	13.76	8.12	69.15	91.03
Order Type 14 (p3-p5)	17.8	39.81	57.61	Order Type 14 (p3-p5)	14.61	6.93	69.15	90.69
Order Type 15 (p4-p5)	19.8	44.10	63.90	Order Type 15 (p4-p5)	14.93	6.29	69.15	90.37
Increased Demand Variation make-to-order Product Cost				**Increased Demand Variation make-to-stock Product Cost**				
P1	12.77	14.57	27.34	P1	11.42	2.84	69.15	83.41
P2	14.17	13.74	27.91	P2	12.82	4.48	69.15	86.45
P3	14.75	16.25	31.00	P3	13.44	4.64	69.15	87.23
P4	18.84	27.85	46.69	P4	14.08	5.92	69.15	89.15
P5	20.84	37.11	57.95	P5	15.78	6.42	69.15	91.35

On the other hand, when the cost is investigated, it is observed that make-to-order strategy responded to orders in a cheaper manner than the make-to-stock strategy. The Increased Demand Variation model with the make-to-order approach provided order and product costs that are very similar if not the same to the original 100 K Demand model with the make-to-order approach. However, when the results of the Increased Demand Variation model with the make-to-stock approach was compared to the original 100 K Demand model with the make-to-stock approach, it is found that the costs with the Increased Demand Variation model with the make-to-stock approach suffered from variations more than 100 K Demand model with the make-to-stock approach. It can be concluded that the average inventory build up creates more cost to the screwdriver company when the variation of the incoming order arrival increases. Therefore, the total costs for orders and products amplify mainly because of the average inventory costs. The results of these experiments can be found in Table 18. Appendix C contains t-tests that lead to draw these conclusions.

Su et al. (2005) studied the higher arrival time variation and their effects on products' customer expected waiting times (lead times). They tested both TP and FP for lead times. They stated that FP is more robust to the arrival time variation than TP. Hence, in this case, it is observed that the queuing theory approach that they used and the simulation approach that this thesis used were in agreement. Su et al. (2005) did not provide results for the consideration of cost in their study; therefore, a comparison was not possible.

4.2.5 Alternate Order Reception Scenarios

Another production strategy tested using a simulation model was the case when the model alternated the way the manufacturer received orders. This thesis refers to this model as One Type of Product in Orders Model. This model assumes that the company receives orders that contain only one type of the product (one of the members of the product family: P1, P2, P3, P4, or P5). How the production process would be affected with respect to customer response times and costs if this order restriction was pushed was the information the simulation model aimed to get. In this case, the company would only have 5 types of orders each including one of the five types of products.

The make-to-stock strategy provided better statistics when the order and product lead times are compared to make-to-order strategy for the two models of One Type of Product in Orders. The lead times of products became within the range of 25%-50% of the make-to-order strategy lead times for the products of the make-to-stock strategy. Although the orders were basic, carrying only one of the five types of screwdriver, compared to the models that were previously studied, the inventory stock created in the make-to-stock strategy was again a very important factor that helped this approach decrease lead times significantly (see Appendix C).

When the lead times of the One Type of Product in Orders model with the make-to-order strategy was compared to the original 100 K Demand model, which has all 15 order combinations as incoming orders, it is found that the original 100 K Demand Model performed better. This can be verified by considering the amount of extra screwdrivers that needed to be produced due to less order types being more loaded in terms of the number of screwdrivers that still meet the same total annual demand. Alternatively, the One Type of Product in Orders model was compared to the original 100 K Demand with the approach of make-to-stock using the lead times performance measure. The One Type of Product in Orders model with the make-to-stock strategy was observed to deliver better results. This can be explained by the presence of variability. Having only one type of product in an order allowed the variability of the system to decrease in general. As a result, the manufacturing system operated more smoothly leading to better lead times of orders and products. Further details of how these results were compared can be found in Appendix C.

On the other hand, when the costs for the One Type of Product in Orders model with make-to-order approach and the One Type of Product in Orders model with make-to-stock approach were compared, it is observed that the One Type of Product in Orders model with make-to-order model is more beneficial for the company. It is also found that as the number of components increase in the products, the differences between the make-to-order and the make-to-stock strategies were getting less pronounced (see Appendix C).

In addition, the total cost values of the original 100 K Demand Model were compared to the One Type of Product in Orders Model with the make-to-order and make-to-stock approach. It was observed that the cost values for both strategies ended up being very similar to each other. The differences of cost values for the same order type with the same scheduling strategy were minimal, changing at most 10% of the cost values of the original 100 K Demand model. Both increases and decreases were found in the One Type of Product in Orders model cost statistics. This can be explained by less variable incoming orders that occurred due to the way the company receives orders. The make-to-stock strategy proves its importance in an environment where the uncertainties of demand and process are considerable. As the system becomes less and less uncertain the use of the make-to-stock strategy loses its importance. Table 19 and 20 provide the statistics collected in these run results. Appendix C contains statistical comparisons with Minitab reports about these observations.

Table 19. The Lead Time Values for the One Type of Product in Orders Model with the Make-to-order and Make-to-stock Strategies

One Type of Product in Orders make-to-order Order Lead Times	Hr.	One Type of Product in Orders make-to-stock Order Lead Times	Hr.
Order Type 1 (p1)	5.3224	Order Type 1 (p1)	3.5595
Order Type 2 (p2)	5.5725	Order Type 2 (p2)	3.3048
Order Type 3 (p3)	6.5748	Order Type 3 (p3)	2.8506
Order Type 4 (p4)	14.003	Order Type 4 (p4)	2.8974
Order Type 5 (p5)	13.435	Order Type 5 (p5)	1.9954
Order Type 6 (p1-p2)	-	Order Type 6 (p1-p2)	-
Order Type 7 (p1-p3)	-	Order Type 7 (p1-p3)	-
Order Type 8 (p1-p4)	-	Order Type 8 (p1-p4)	-
Order Type 9 (p1-p5)	-	Order Type 9 (p1-p5)	-
Order Type 10 (p2-p3)	-	Order Type 10 (p2-p3)	-
Order Type 11 (p2-p4)	-	Order Type 11 (p2-p4)	-
Order Type 12 (p2-p5)	-	Order Type 12 (p2-p5)	-
Order Type 13 (p3-p4)	-	Order Type 13 (p3-p4)	-
Order Type 14 (p3-p5)	-	Order Type 14 (p3-p5)	-
Order Type 15 (p4-p5)	-	Order Type 15 (p4-p5)	-
One Type of Product in Orders make-to-order Product Lead Times		**One Type of Product in Orders make-to-stock Product Lead Times**	
P1	3.3218	P1	1.9503
P2	3.4958	P2	1.8198
P3	4.4741	P3	1.4525
P4	9.0243	P4	1.8862
P5	7.1754	P5	1.1531

Table 20. The Cost Values for the One Type of Product in Orders Model with the Make-to-order and Make-to-stock Strategies

One Type of Product in Orders make-to-order Order Cost	**$**	**$**	**$**	**One Type of Product in Orders make-to-stock Order Cost**	**$**	**$**	**$**	**$**
	Fixed Cost	**WIP Inventory Cost**	**Total Cost**		**Fixed Cost**	**WIP Inventory Cost**	**Average Inventory Holding Cost**	**Total Cost**
Order Type 1 (p1)	12.8	13.63	26.43	Order Type 1 (p1)	11.42	8.13	69.15	88.70
Order Type 2 (p2)	14.2	15.83	30.03	Order Type 2 (p2)	12.82	8.47	69.15	90.44
Order Type 3 (p3)	14.8	19.46	34.26	Order Type 3 (p3)	13.44	7.66	69.15	90.25
Order Type 4 (p4)	18.8	52.65	71.45	Order Type 4 (p4)	14.08	8.16	69.15	91.39
Order Type 5 (p5)	20.8	55.89	76.69	Order Type 5 (p5)	15.78	6.30	69.15	91.23
Order Type 6 (p1-p2)	13.5	-	-	Order Type 6 (p1-p2)	12.12	-	69.15	-
Order Type 7 (p1-p3)	13.8	-	-	Order Type 7 (p1-p3)	12.43	-	69.15	-
Order Type 8 (p1-p4)	15.8	-	-	Order Type 8 (p1-p4)	12.75	-	69.15	-
Order Type 9 (p1-p5)	20.8	-	-	Order Type 9 (p1-p5)	15.78	-	69.15	-
Order Type 10 (p2-p3)	14.5	-	-	Order Type 10 (p2-p3)	13.13	-	69.15	-
Order Type 11 (p2-p4)	16.5	-	-	Order Type 11 (p2-p4)	13.45	-	69.15	-
Order Type 12 (p2-p5)	17.5	-	-	Order Type 12 (p2-p5)	14.3	-	69.15	-
Order Type 13 (p3-p4)	16.8	-	-	Order Type 13 (p3-p4)	13.76	-	69.15	-
Order Type 14 (p3-p5)	17.8	-	-	Order Type 14 (p3-p5)	14.61	-	69.15	-
Order Type 15 (p4-p5)	19.8	-	-	Order Type 15 (p4-p5)	14.93	-	69.15	-
One Type of Product in Orders make-to-order Product Cost				**One Type of Product in Orders make-to-stock Product Cost**				
P1	12.77	8.48	21.25	P1	11.42	4.45	69.15	85.02
P2	14.17	9.91	24.08	P2	12.82	4.67	69.15	86.64
P3	14.75	13.20	27.95	P3	13.44	3.90	69.15	86.49
P4	18.84	34.00	52.84	P4	14.08	5.31	69.15	88.54
P5	20.84	29.91	50.75	P5	15.78	3.64	69.15	88.57

Su et al. (2005) discussed this issue only from the perspective of how higher and lower arrival time variations would affect the system. In addition, as mentioned previously, they only analyzed the TP and FP alternatives from an expected waiting times, lead times, point of view. They concluded their discussion by proving the robustness of the FP, make-to-stock strategy, in larger arrival time variations.

4.2.6 Alternate Product Family Designs

In addition, the product family alternatives were tested using the simulation study. Alternative designs of the product family with three members and four members instead of the five types (P1, P2, P3, P4, and P5) were investigated. The feasibility of a decision to produce three and four types of products is studied. Since the functionality of the products increased within the product family P1 to P5, it was decided to cut the production of P1 and assign all orders including P1 to P2 for a test of product family with four members and to cut the production of P1 and P2 and assign all orders including P1 and P2 to P3. P2 was capable of meeting the customers' requirements in terms of functionality when it was offered to the customer instead of P1. Likewise, P3 was capable of meeting the customers' requirements in terms of functionality when it was offered to the customers instead of P1 and P2. The more capable products cost more than the less capable ones. The study is to demonstrate whether cutting the production of P1 and P2 could be justified in terms of lead times and costs. The thesis calls these models as Three Types of Product in Product Family and Four Types of Product in Product Family.

When the Four Types of Product in Product Family was run with make-to-stock strategy and with the make-to-order strategy, it was found that the make-to-stock strategy performed better than make-to-order strategy once again in terms of lead times of order and products. Having the common components ready for production all the time regardless of the order arrivals helped the company decrease the lead time significantly also with Four Types of Product in Product Family like all other alternative product family design and assembly scheduling strategy conditions studied.

Further, the statistics of the Four Types of Product Family model was compared to the statistics of the original 100 K Demand Model. The lead times of the Four Types of Product Family Model with the make-to-order approach was generally increased for the orders containing screwdriver P2 compared to the original 100 K Demand for the make-to-order option. However, these increases remained as a low percentage of the original 100 K Demand Model with make-to-order Model lead times. These small lead time increases could be explained by the additional components that were produced in the P2 instead of the components of P1. However, since screwdriver P1 and screwdriver P2 components are similar to each other, the effect of having only four product types in the screwdriver family remained small. The t-tests supported these small impacts with p-values yielding insignificance (see Appendix C for further details). Furthermore, the make-to-stock approach of the Four Types of Product Family Model was also compared to the make-to-stock approach of the original 100 K Demand Model. It was found that the order and product lead times were still not significantly affected. The make-to-stock approach produced certain order types, especially the ones carrying P2 in them, in slightly longer lead time than the original 100 K Demand Model produced. However, because these differences in the lead times were small, they are considered to be negligible in the long run. The confidence intervals were justifying these small differences. This can be explained by the reduced variety in the system due to a smaller number of members in the product family. Although the manufacturing facility had to produce screwdrivers that are more complicated than what their customers demand, they also decrease the variability in their system. This helps them improve their production flow.

From a cost perspective, the model using the make-to-order approach of the Four Types of Product in Product Family Model was still performing better than the model using the make-to-stock approach. Once more, the effect of average inventory cost became a major factor in the cost calculations. This led the make-to-stock strategy to become a costly one for the screwdriver manufacturing company (see Appendix C for t-tests).

When the cost values that the Four Types of Product in Product Family Model produced were compared to the cost values that the original 100 K Demand Model produced, it is observed that these cost values remained the same when they both used the make-to-order strategy. The raw material cost difference between P1 and P2 is very low considering the total cost values. The flow of production did not seem to suffer replacing the P1's in orders with P2's cost wise. In addition, the comparison of the cost values that the Four Types of Product in Product Family Model produced were compared to the cost values that the original 100 K Demand Model produced using the make-to-stock strategy was performed. It was noticed that the exclusion of P1 only increased the cost of the order types that included P2 and P3. However, this increase was not considered significant according to the statistical analysis that was used for this system. The findings of the Three Types of Product in Product Family Model were also similar to the findings of Four Types of Product in Product Family Model. The importance of producing the first two types of products could be argued using the same reasoning. In short, it can be concluded that the simulation models created for both assembly scheduling strategies responded well to the idea of decreasing the number of screwdrivers in the product family. The company should consider removing their most basic component from market. Tables 21, 22, 23, and 24 contain the results of these experiments.

Table 21. The Lead Time Values for the Four Types of Product in Product Family Model

with the Make-to-order and Make-to-stock Strategies

Four Types of Product in Product Family make-to-order Order Lead Times	**Hr.**	**Four Types of Product in Product Family make-to-stock Order Lead Times**	**Hr.**
Order Type 1 (p1)	-	Order Type 1 (p1)	-
Order Type 2 (p2)	4.3808	Order Type 2 (p2)	3.0316
Order Type 3 (p3)	5.3732	Order Type 3 (p3)	2.7188
Order Type 4 (p4)	14.457	Order Type 4 (p4)	2.1904
Order Type 5 (p5)	18.145	Order Type 5 (p5)	2.6822
Order Type 6 (p1-p2)	8.2266	Order Type 6 (p1-p2)	3.4494
Order Type 7 (p1-p3)	5.8881	Order Type 7 (p1-p3)	2.8520
Order Type 8 (p1-p4)	11.846	Order Type 8 (p1-p4)	2.8774
Order Type 9 (p1-p5)	11.178	Order Type 9 (p1-p5)	3.3543
Order Type 10 (p2-p3)	7.4973	Order Type 10 (p2-p3)	2.8400
Order Type 11 (p2-p4)	16.213	Order Type 11 (p2-p4)	4.1663
Order Type 12 (p2-p5)	16.374	Order Type 12 (p2-p5)	2.1817
Order Type 13 (p3-p4)	16.079	Order Type 13 (p3-p4)	3.5976
Order Type 14 (p3-p5)	7.1395	Order Type 14 (p3-p5)	2.9347
Order Type 15 (p4-p5)	15.825	Order Type 15 (p4-p5)	2.2765
Four Types of Product in Product Family make-to-order Product Lead Times		**Four Types of Product in Product Family make-to-stock Product Lead Times**	
P1	-	P1	-
P2	5.1846	P2	1.7267
P3	4.8355	P3	1.7267
P4	10.823	P4	2.0668
P5	10.946	P5	2.1360

Table 22. The Cost Values for the Four Types of Product in Product Family Model

with the Make-to-order and Make-to-stock Strategies

Four Types of Product in Product Family make-to-order Order Cost	**$**	**$**	**$**	**Four Types of Product in Product Family make-to-stock Order Cost**	**$**	**$**	**$**	**$**
	Fixed Cost	**WIP Inventory Cost**	**Total Cost**		**Fixed Cost**	**WIP Inventory Cost**	**Average Inventory Holding Cost**	**Total Cost**
Order Type 1 (p1)	-	-	-	Order Type 1 (p1)	-	-	-	-
Order Type 2 (p2)	14.2	12.44	26.64	Order Type 2 (p2)	12.82	7.77	69.15	89.74
Order Type 3 (p3)	14.8	15.90	30.70	Order Type 3 (p3)	13.44	7.31	69.15	89.90
Order Type 4 (p4)	18.8	54.36	73.16	Order Type 4 (p4)	14.08	6.17	69.15	89.40
Order Type 5 (p5)	20.8	75.48	96.28	Order Type 5 (p5)	15.78	8.47	69.15	93.40
Order Type 6 (p1-p2)	13.5	22.21	35.71	Order Type 6 (p1-p2)	12.12	8.36	69.15	89.63
Order Type 7 (p1-p3)	13.8	16.25	30.05	Order Type 7 (p1-p3)	12.43	7.09	69.15	88.67
Order Type 8 (p1-p4)	15.8	37.43	53.23	Order Type 8 (p1-p4)	12.75	7.34	69.15	89.24
Order Type 9 (p1-p5)	20.8	46.50	67.30	Order Type 9 (p1-p5)	15.78	10.59	69.15	95.52
Order Type 10 (p2-p3)	14.5	21.74	36.24	Order Type 10 (p2-p3)	13.13	7.46	69.15	89.74
Order Type 11 (p2-p4)	16.5	53.50	70.00	Order Type 11 (p2-p4)	13.45	11.21	69.15	93.81
Order Type 12 (p2-p5)	17.5	57.31	74.81	Order Type 12 (p2-p5)	14.3	6.24	69.15	89.69
Order Type 13 (p3-p4)	16.8	54.03	70.83	Order Type 13 (p3-p4)	13.76	9.90	69.15	92.81
Order Type 14 (p3-p5)	17.8	25.42	43.22	Order Type 14 (p3-p5)	14.61	8.58	69.15	92.34
Order Type 15 (p4-p5)	19.8	62.67	82.47	Order Type 15 (p4-p5)	14.93	6.80	69.15	90.88
Four Types of Product in Product Family make-to-order Product Cost				**Four Types of Product in Product Family make-to-stock Product Cost**				
P1	-	-	-	P1	-	-	-	-
P2	14.17	14.69	28.86	P2	12.82	4.43	69.15	86.40
P3	14.75	14.26	29.01	P3	13.44	4.64	69.15	87.23
P4	18.84	40.78	59.62	P4	14.08	5.82	69.15	89.05
P5	20.84	45.62	66.46	P5	15.78	6.74	69.15	91.67

Table 23. The Lead Time Values for the Three Types of Product in Product Family Model with the Make-to-order and Make-to-stock Strategies

Three Types of Screwdriver Order Lead Times	Hr.	Three Types of Screwdriver Order Lead Times	Hr.
Order Type 1 (p1)	-	Order Type 1 (p1)	-
Order Type 2 (p2)	-	Order Type 2 (p2)	-
Order Type 3 (p3)	5.3732	Order Type 3 (p3)	2.7953
Order Type 4 (p4)	14.457	Order Type 4 (p4)	1.9886
Order Type 5 (p5)	18.145	Order Type 5 (p5)	2.7014
Order Type 6 (p1-p2)	8.2266	Order Type 6 (p1-p2)	3.4528
Order Type 7 (p1-p3)	5.8881	Order Type 7 (p1-p3)	2.8896
Order Type 8 (p1-p4)	11.846	Order Type 8 (p1-p4)	2.3042
Order Type 9 (p1-p5)	11.178	Order Type 9 (p1-p5)	2.4921
Order Type 10 (p2-p3)	7.4973	Order Type 10 (p2-p3)	3.4014
Order Type 11 (p2-p4)	16.213	Order Type 11 (p2-p4)	4.0382
Order Type 12 (p2-p5)	16.374	Order Type 12 (p2-p5)	2.5523
Order Type 13 (p3-p4)	16.079	Order Type 13 (p3-p4)	2.6834
Order Type 14 (p3-p5)	7.1395	Order Type 14 (p3-p5)	2.4407
Order Type 15 (p4-p5)	15.825	Order Type 15 (p4-p5)	2.6733
Three Types of Screwdriver Product Lead Times		**Three Types of Screwdriver -Stock Product Lead Times**	
P1	-	P1	-
P2	-	P2	-
P3	5.0718	P3	1.5612
P4	10.823	P4	1.8031
P5	10.946	P5	1.9822

Table 24. The Cost Values for the Three Types of Product in Product Family Model

with the Make-to-order and Make-to-stock Strategies

Three Types of Product in Product Family **make-to-order Order Cost**	**$ Fixed Cost**	**$ WIP Inven. Cost**	**$ Total Cost**	**Three Types of Product** In Product Family **make-to-stock Order Cost**	**$ Fixed Cost**	**$ WIP Inven. Cost**	**$ Ave. Inven. Holding Cost**	**$ Total Cost**
Order Type 1 (p1)	-	-	-	Order Type 1 (p1)	-	-	-	-
Order Type 2 (p2)	-	-	-	Order Type 2 (p2)	-	-	-	-
Order Type 3 (p3)	14.8	15.90	30.70	Order Type 3 (p3)	13.44	7.51	69.15	90.10
Order Type 4 (p4)	18.8	54.36	73.16	Order Type 4 (p4)	14.08	5.60	69.15	88.83
Order Type 5 (p5)	20.8	75.48	96.28	Order Type 5 (p5)	15.78	8.53	69.15	93.46
Order Type 6 (p1-p2)	13.5	22.21	35.71	Order Type 6 (p1-p2)	12.12	8.37	69.15	89.64
Order Type 7 (p1-p3)	13.8	16.25	30.05	Order Type 7 (p1-p3)	12.43	7.18	69.15	88.76
Order Type 8 (p1-p4)	15.8	37.43	53.23	Order Type 8 (p1-p4)	12.75	5.88	69.15	87.78
Order Type 9 (p1-p5)	20.8	46.50	67.30	Order Type 9 (p1-p5)	15.78	7.87	69.15	92.80
Order Type 10 (p2-p3)	14.5	21.74	36.24	Order Type 10 (p2-p3)	13.13	8.93	69.15	91.21
Order Type 11 (p2-p4)	16.5	53.50	70.00	Order Type 11 (p2-p4)	13.45	10.86	69.15	93.46
Order Type 12 (p2-p5)	17.5	57.31	74.81	Order Type 12 (p2-p5)	14.3	7.30	69.15	90.75
Order Type 13 (p3-p4)	16.8	54.03	70.83	Order Type 13 (p3-p4)	13.76	7.38	69.15	90.29
Order Type 14 (p3-p5)	17.8	25.42	43.22	Order Type 14 (p3-p5)	14.61	7.13	69.15	90.89
Order Type 15 (p4-p5)	19.8	62.67	82.47	Order Type 15 (p4-p5)	14.93	7.98	69.15	92.06
Three Types of Product in Product Family make-to-order Product Cost				**Three Types of Product in Product Family make-to-stock Product Cost**				
P1	-	-	-	P1	-	-	-	-
P2	-	-	-	P2	-	-	-	-
P3	14.75	14.96	29.71	P3	13.44	4.20	69.15	86.79
P4	18.84	40.78	59.62	P4	14.08	5.08	69.15	88.31
P5	20.84	45.62	66.46	P5	15.78	6.26	69.15	91.19

Su et al. (2005) studied alternative product designs with queuing theory for improvement for their selected performance measures, customer response times and total product costs. They worked with different percentage levels of common components in their product family and different numbers of products in the product family. They stated that as the common components increases the expected lead time for the make-to-stock strategy, the lead times for the orders decreases. This is straightforward considering that make-to-stock strategy allows components to be ready in inventory before they are assembled to the products when the orders arrive and shorten the total lead time. Having more common components ready would help to company decrease the lead time. In addition, they also studied the number of product family members in the product family design. They stated that as the number of products in the family increases, both cost and total lead time of products would increase with the make-to-stock strategy. More products in the family would increase the inventory required to be held by the company and that would increase the total costs because of average inventory cost. Lead times would also increase because more products in the family would increase the variability in the products and lead to a decrease of the number of common components. As a result, the company that is not able to take advantage of creating inventory of common components would suffer in terms of lead times because their production would become more and more customer oriented. However, if the make-to-order strategy is used by the company, increasing the number of products in the family would not increase the costs or the lead times. Since in this (TP) strategy, waits for the order arrivals for production, more products in the product family would not affect the production operations of the company. In light of these observations, Su et al. (2005) concluded that manufacturing companies should consider improving their products rather than their processes. This thesis investigated the same issue by alternating the number of the product in the product family. In addition to simulation model representing manufacturing systems producing five screwdrivers, simulation models that manufacture three or four screwdrivers were created. This simulation study found that the system would improve when the number of screwdriver in the product family was lowered because the reflection of these decreases in the number of products in the product family to the dependent variables, lead times and costs, were insignificant.

4.2.7 Alternate Product Family Designs and Order Reception Scenarios

Finally, the simulation study of this thesis investigated how the system would react to the case when there are four types of products in the product family and orders include only one type of product. There would only be four different types of orders including one of the P2, P3, P4, and P5. This model is called the Four Types of Product in Product Family, One Product per Order Model. With the order and product lead times of the Four Types of Product in Product Family, One Product per Order model, the make-to-stock strategy performed better results than make-to-order strategy. Having the common components for four product types instead of five did not affect the make-to-stock strategy from decreasing the lead times using previously produced and stocked components in the moment of order arrival. On the other hand, when the Four Types of Product in Product Family, One Product per Order Model was designed for investigation of assembly scheduling strategies, make-to-order vs. make-to-stock strategies, the total costs of products suffered from average inventory costs nearly doubling the total costs for higher common component percentage products. The more complicated screwdriver received the least harm from average inventory cost because of its low percentage of common components in the products.

When the make-to-stock and the make-to-order strategies' results of the Four Types of Product in Product Family, One Product per Order model was compared with the make-to-stock and the make-to-order strategies' results of the original 100 K Demand model, it is observed that the Four Types of Product in Product Family, One Product per Order model suffered in terms of order and product lead times in the make-to-stock strategy. Both product and order lead times increased in the range of 15% to 50% percent depending on the orders and products. The lead times of Four Types of Product in Product

Family, One Product per Order Model with make-to-stock strategy are affected by the change of order reception and the exclusion of P1 from the product family because all the demand for the P1 gets assigned to Order Type 2 and causes a bottleneck in the system. Although the variability in the system is the lowest in all alternative production methods tested for this thesis, the increase in Order Type 2 demand causes a lack of balance to the system. However, when the lead times of orders and products of Four Types of Product in Product Family, One Product per Order Model and the original 100 K Demand model was compared, it was found that these statistics were not significantly affected. Therefore, it can be concluded that eliminating the production of P1 is an option that the company needs to consider in order to simplify their product line.

On the other hand, from a cost perspective, the production of the make-to-stock strategy of the Four Types of Product in Product Family, One Product per Order model was more costly to the firm compared to the make-to-order strategy of the Four Types of Product in Product Family, One Product per Order model both in order and product costs. This again could be explained by the average inventory holding cost that is brought to the system when the make-to-stock strategy is applied. Even in a less variable environment created by the less number of products in the product family and the alternative order reception, the average inventory cost remained to be the main driver of total cost for the screwdriver facility. When the cost results of the Four Types of Product in Product Family, One Product per Order model were compared to the original 100 K Demand model, it was found that for the make-to-order strategy, the costs were not significantly affected. Since the make-to-order strategy only triggers production with the incoming orders, the production was not impacted by the smaller number of products in the product family or the less variability in the order reception. Furthermore, when the cost results of the Four Types of Product in Product Family, One Product per Order Model with the make-to-stock strategy were compared to the original 100 K Demand model with the make-to-stock strategy, it was noticed that the costs remained indifferent except P2, which fulfills the demand for P1. The costs of

P2 became very high because it also carried the customer's demand for P1. This can be explained by the excessive amount of demand for P2 forcing the inventory system run out of common components and creating backlog for the whole system. The statistics of these runs can be found in Tables 25 and 26.

Table 25. The Lead Time Values for the Four Types of Product in Product Family One Product per Order Model

with the Make-to-order and Make-to-stock Strategies

Four Types of Product in Product Family One Product per Order make-to-order Order Lead Times	**Hr.**	**Four Types of Product in Product Family One Product per Order make-to-stock Order Lead Times**	**Hr.**
Order Type 1 (p1)	-	Order Type 1 (p1)	-
Order Type 2 (p2)	5.5725	Order Type 2 (p2)	3.3566
Order Type 3 (p3)	6.5748	Order Type 3 (p3)	2.8365
Order Type 4 (p4)	14.003	Order Type 4 (p4)	2.8056
Order Type 5 (p5)	13.435	Order Type 5 (p5)	1.9998
Four Types of Product in Product Family One Product per Order make-to-order Product Lead Times		**Four Types of Product in Product Family One Product per Order make-to-stock Product Lead Times**	
P1	-	P1	-
P2	3.4016	P2	1.8783
P3	4.4741	P3	1.4455
P4	9.0243	P4	1.7567
P5	7.1754	P5	1.1558

Table 26. The Cost Values for the Four Types of Product in Product Family One Product per Order Model with the Make-to-order and Make-to-stock Strategies

Four Types of Product in Product Family One Type of Product in Orders make-to-order Order Cost	$	$	$	Four Types of Product in Product Family One Type of Product in Orders make-to-stock Order Cost	$	$	$	$
	Fixed Cost	WIP Inventory Cost	Total Cost		Fixed Cost	WIP Inventory Cost	Average Inventory Holding Cost	Total Cost
Order Type 1 (p1)	-	-	-	Order Type 1 (p1)	-	-	-	-
Order Type 2 (p2)	14.2	15.83	30.03	Order Type 2 (p2)	12.82	8.61	69.15	90.58
Order Type 3 (p3)	14.8	19.46	34.26	Order Type 3 (p3)	13.44	7.62	69.15	90.21
Order Type 4 (p4)	18.8	52.65	71.45	Order Type 4 (p4)	14.08	7.90	69.15	91.13
Order Type 5 (p5)	20.8	55.89	76.69	Order Type 5 (p5)	15.78	6.31	69.15	91.24
Four Types of Product in Product Family One Type of Product in Orders make-to-order Product Cost				Four Types of Product in Product Family One Type of Product in Orders make-to-stock Product Cost				
P1	-	-	-	P1	-	-	-	-
P2	14.17	9.64	23.81	P2	13.7	8.4	69.15	90.37
P3	14.75	13.20	27.95	P3	14.3	8.1	69.15	90.69
P4	18.84	34.00	52.84	P4	18.4	16.9	69.15	100.13
P5	20.84	29.91	50.75	P5	20.1	14.7	69.15	99.63

Tables 27 and 28 summarize the results of experiments for models Three Types of Screwdriver One type of Screwdriver in Order with make-to-order and make-to-stock approaches. These results were compared using pair-wise t-tests, which can be found in Appendix C. Similar results once again were received to the Four Types of Screwdriver One type of Screwdriver in Order model.

Su et al. (2005) did not go into much detail about studying both incoming order arrivals and the product family design alternatives at the same time. They studied the percentage of common components and the number of products in the family. They concluded that increasing the percentage of common components and decreasing the number of products in the family improves the expected customer waiting times, the lead times, in other words. However, they also mentioned that lowering the variation in the order arrivals does not significantly improve the lead times in the make-to-stock strategy. This final comment was also confirmed by this thesis' simulation approach.

Table 27. The Lead Time Values for the Three Types of Product in Product Family Model One Type of Screwdriver in Order with the Make-to-order and Make-to-stock Strategies

Three Types of Screwdriver One type of Screwdriver in Order make-to-order Order Lead Times	**Hr.**	**Three Types of Screwdriver One type of Screwdriver in Order make-to-stock Order Lead Times**	**Hr.**
Order Type 1 (p1)	-	Order Type 1 (p1)	-
Order Type 2 (p2)	-	Order Type 2 (p2)	-
Order Type 3 (p3)	6.5748	Order Type 3 (p3)	2.9415
Order Type 4 (p4)	14.003	Order Type 4 (p4)	2.672
Order Type 5 (p5)	16.182	Order Type 5 (p5)	1.8604
Three Types of Screwdriver One type of Screwdriver in Order make-to-order Product Lead Times		**Three Types of Screwdriver One type of Screwdriver in Order make-to-stock Product Lead Times**	
P1	-	P1	-
P2	-	P2	-
P3	3.7869	P3	1.5985
P4	9.0243	P4	1.639
P5	7.1754	P5	1.0546

Table 28. The Cost Values for the Three Types of Product in Product Family Model One Type of Screwdriver in Order with the Make-to-order and Make-to-stock Strategies

Three Types of Product in Product Family One Type of Product in Orders make-to-order Order Cost	$	$	$	Three Types of Product in Product Family One Type of Product in Orders make-to-stock Order Cost	$	$	$	$
	Fixed Cost	WIP Inv. Cost	Total Cost		Fixed Cost	WIP Inv. Cost	Ave. Inv. Holding Cost	Total Cost
Order Type 1 (p1)	-	-	-	Order Type 1 (p1)	-	-	-	-
Order Type 2 (p2)	-	-	-	Order Type 2 (p2)	-	-	-	-
Order Type 3 (p3)	14.8	19.46	34.26	Order Type 3 (p3)	13.44	7.91	69.15	90.5
Order Type 4 (p4)	18.8	52.65	71.45	Order Type 4 (p4)	14.08	7.52	69.15	90.75
Order Type 5 (p5)	20.8	67.32	88.12	Order Type 5 (p5)	15.78	5.87	69.15	90.8
Three Types of Product in Product Family One Type of Product in Orders make-to-order Product Cost				**Three Types of Product in Product Family One Type of Product in Orders make-to-stock Product Cost**				
P1	-	-	-	P1	-	-	-	-
P2	-	-	-	P2	-	-	-	-
P3	14.75	11.17	25.92	P3	14.3	4.57	69.15	88.02
P4	18.84	34	52.84	P4	18.4	6.03	69.15	93.58
P5	20.84	29.91	50.75	P5	20.1	4.24	69.15	93.49

4.3 Summary

The results discussed in the previous two sections are achieved using outcomes of the simulation models representing a screwdriver manufacturing system with alternative production strategies. This research is targeted to be a guide for factory managers in their decisions about how to increase their profitability and customer satisfaction. It mainly investigates the impacts of alternative product family design strategies and postponement strategies under certain demand changes and shop floor modifications. The response of the system is observed using the two dependent variables, lead times and product costs.

First, a full factorial design for strategies that are considered important is completed and analyzed using MANOVA. This study showed that the independent variables, demand level and postponement strategies, were causing significant differences on the two dependent variables, lead times and costs. On the other hand, the MANOVA analysis for the product family design alternatives concluded that the number of products in the product family for the studied screwdriver family was not a main factor improving the system performance. MANOVA found that the alternative design strategies did not make a significant impact on the system in terms of the two dependent variables (lead times and cost) studied.

Another conclusion that the study found was that, in general, the form postponement strategy (make-to-stock) was costly to the firm. However, it helped company to improve customer satisfaction by decreasing the lead time. Producing the selected common components before the order arrivals and stocking them as unfinished inventory would be a good strategy for a company that would like to reach the customers who are willing to pay more money for products despite their possible higher costs.

Individual comparisons of the impacts of the selected scenario changes were performed on the system for further investigation using pair-wise comparisons that t-tests provided. These comparisons were useful because they provided insights about in which direction the changes affect the system. In general, similar results were observed to those of MANOVA completed for this study. These individual comparisons were compared to the findings of Su et al. (2005).

The results of the stochastic study from Su et al. (2005) can be summarized in six main points: alternate demand scenarios, alternate number of machining, alternate variation in demand scenarios of inter-arrival time, alternate product family design strategies, and alternate product family designs and order reception scenarios. The findings of Su et al. (2005) and this thesis were compared in the previous sections in detail. They are also summarized in table 29.

Table 29. Summary of Individual Comparisons

Stochastic Study vs. Simulation Study	Stochastic Study	Simulation Study
alternate demand systems (higher utilization levels in the stochastic study and higher demand in the simulation study)	It was found that that the make-to-stock strategy is more robust against the alternative scenarios in terms of the costs and lead times of the production.	The make-to-stock simulation model takes advantage of the inventory and decreases the impact of the alternate demand on the system.
alternate capacity (number of resources differing in both studies)	Additional resources did not impact the lead times and costs for the make-to-stock model.	Lead times were impacted through additional resources. Lower lead time values were achieved. Impact on make-to-stock was insignificant.
demand variability (the inter-arrival times of order differing in both studies)	The study stated that FP is more robust to the arrival time variation than TP.	Variability was checked for both make-to-order and make-to-stock. Make-to-stock was not impacted as much as make-to-order.
larger order arrival variation (reception of order differing in the simulation study)	The robustness of the FP, make-to-stock strategy, was proved in larger arrival time variations	When the variability is differed, make-to-stock provided more stable results than make-to-order.
alternate product family design (number of products in the family differing in both studies)	Su et al. (2005) concluded that manufacturing companies should consider improving their products rather than their processes.	The impact of the number of screwdriver in the product family in terms of lead times and costs, were insignificant.
alternate product family design and demand variability (number of products in the family differing in both studies and reception of order differing in the simulation study)	Increasing the percentage of common components improves the lead times. Lowering the variation in the order arrivals does not significantly improve the lead times in the make-to-stock strategy.	Product family alternatives did not create significant impact. However, variations impacted the system significantly.

The findings of this work and Su et al. (2005) were parallel when alternate demand scenarios, alternate number of machining, alternate variation in demand scenarios of inter-arrival time, and alternate product family designs and order reception scenarios were analyzed. These changes impacted the system in the same direction in terms of the two performance measures, lead times and costs, investigated by the two studies.

However, when alternate product family design strategies were considered and tested, the two studies concluded different results. The simulation study proved that the system has not improved in terms of both lead times and costs after the number of products in the product family investigated is decreased. The t-test analysis found the changes in the performance measures to be insignificant when alternated product family strategies were tested by the simulation runs. The nature of the product studied had an important role in this finding. Although the number of components in screwdrivers P1-P5 was varying from 17 to 43, when these components are studied more closely, it is found that total process times were not largely different among the product family alternatives. Table 30 analyzes these total process times in comparison with the total lead times and costs created by the 100 K Demand model make-to-stock strategy. Alternate product family options of four products in the family and three products in the family are created by the elimination of P1 and P1 and P2 from the system. The demand related to these products is replaced by P2 and P3 respectively in order to modify the product family design. The elimination of P4 and P5 was not considered because of their functionalities. In terms of their functionalities, the company absolutely needed to have those two products in the market.

It is observed that the process times of P1, P2 and P3 are numerically close to each other. When the total process times and total lead times are considered, it is observed that the product family alternative with four products in the family offers an improvement of a maximum 3.5% of the total process time. Meanwhile, the product family of three products in the family offers an improvement of a maximum 5% of the total process time. As a result, because these percentages are minor, the elimination of P1 and P2 is not as effective in terms of the average lead times and costs of the system as observed in the table 30.

Table 30. Total Process Times vs. Product Lead Times and Costs

	Total Process Times (sec.)	Product Lead Times (hr.)	Product Costs ($)
P1	436	1.3272	83.6
P2	451	1.6403	86.18
P3	458	1.7495	87.29
P4	1817	1.8469	88.43
P5	2095	2.0893	91.52

Therefore, it can be concluded that the nature of the product is important when product family design strategies are suggested as improvements for a system. For alternate product family designs to impact the system, large variations of total process times of components are necessary. In the screwdriver manufacturing system studied, scheduling decisions were more critical than product family design decisions as such decisions more drastically impacted the performance measures, lead times and costs.

Chapter 5

Summary, Limitations of the Research

And Future Work

5.1 Summary

The issues of product family design and assembly scheduling have been investigated in this thesis using computer simulation in order to incorporate a randomness that is close to the one that companies face in the real world. The use of simulation study for the investigation of product family design, alternative resource availability, potential demand fluctuations, postponement strategies and the used performance measures was not found in the literature review. From this point of view, this work is original, and hence contributes to the state-of the art. In the simulation studies, it was found that the make-to-order strategy is the one that mostly delivered the product in a less expensive manner for the screwdriver family. However, it was not very helpful with quick customer response times. On the other hand, it was also found that the make-to-stock strategy, in other words form postponement, as indicated in the literature review, is usually better when the performance measure is the order or product lead times. In spite of this customer response time advantage, because of the introduction of inventory cost to the cost calculations, the make-to-stock strategy is costly in general. Companies facing the decision between the two assembly scheduling strategies should determine their priorities and optimize their operations accordingly.

In addition, this research shows how possible changes in the operating environment would be handled by the manufacturing system. Changes in demand such as increases in demand variability and changes in demand level were investigated. The reactions of time postponement and form postponement strategies to these changes were studied. Increased demands proved the significance of the make-to-stock strategy with improved performance at the lead time level without significantly affecting the cost. Also, the variability of the demand was reduced by the strategy change in the order reception. Less variable demand helped the system improving lead times and costs.

Companies in today's world should keep in mind that they try to survive in a very dynamic environment. Hence, when making decisions, they need to take into account that they might face changes in years to come. Besides, there is always the chance of improving their facilities either with additional resources or resources that are more capable thanks to technological developments. The improvement in the lead times with additional machining is evident. However, in this analysis, investment costs of these machines need to be taken into account to the full picture especially from the cost perspective. Making decisions like determining the assembly scheduling strategies and establishing the product family designs may require these sorts of presumptions.

In addition, the design of the product family is important in the decision set of manufacturers in competition. The nature of the product and the component selection is important when better performance is targeted through product family design improvements.

This study aims to have product designers and process engineers take a closer look at how assembly sequencing is impacted by product family design and manufacturing uncertainties. As mentioned earlier in the literature review, product designers should become more informed about the potential design impacts on the manufacturing complexity. The experiments conducted in this study allow product designers and process engineers to consider manufacturing intricacies while deciding the number of product variants within a family and the assembly scheduling method to be selected for it -- time postponement or form postponement.

5.2 Limitations of the Research

This research had limitations. First, the information presented in Park (2005) was not complete to create the simulation model with all the details. Certain assumptions needed to be made so the flow of the system would be established. All components of the screwdrivers were investigated carefully for this research when the bills of materials were being created. At that time, the selection of common and variant components was done to balance of the system. However, having this information provided from the manufacturing company would be the ideal situation. Finally, since the simulation model required creation of thousands of entities at any moment of the experiments, the hardware used to handle this model was a limitation. The number of entities that the system can handle at a time limited this study in terms of the duration of the runs. However, the 500 hours of simulation duration was good enough as it reached the steady state for the performance measures investigated.

5.3 Future Work

The study of the issue of product family design and assembly scheduling can be extended by analyzing the effects of these decisions for the whole supply chain rather than a sole manufacturing system. This would give a better understanding of the system with all the outsourced productions included in the analysis. In addition, variability would be better reflected to the study when all supply chain units are taken into account. Garg and Tang (1997) stated that further analysis on the postponement strategies with the incorporation of different costs such as design cost, transportation cost, and distribution cost is needed.

Different product families can be studied to check for the same alternative strategies and conditions. The screwdrivers investigated in this study had many components. Different number of components and different fractions of common components to uncommon components could lead to new findings. In addition, the total process times of components are important. The feasibility of product family design alternatives depend on the total process times of the component of the product studied. Highly variant total process times provide more opportunities to improve the performance measures such as lead times and costs of the system through product family design decisions. These findings would benefit manufacturers.

In addition, the preferences of companies can be studied to a greater extent. This would be helpful in the process of making the decision whether the main competitive priority of an individual company is the cost or the customer response time. In most cases, a combination of the two priorities might be necessary. This would become a multi-objective problem. Studying the customer demand, companies can find the point where they can meet the cost and customer response time.

References

Alizon, F., Shooter, S. B., Simpson, T. W., 2006. Reuse of Manufacturing Knowledge to Facilitate Platform-Based Product Realization. Transactions of ASME, Journal of Computing and Information Science in Engineering, 6 (2), 170-178.

Baldwin, C. Y., Clark, K. B., 2000. Design Rules: The Power of Modularity. MIT Press. Cambridge, MA.

Bi, Z. M. and Zhang, W. J., 2001. Modularity Technology in Manufacturing: Taxonomy and Issues. International Journal Advanced Manufacturing Technology, 18 (5), 381-390.

Bruccoleri, M., Amico, M., Perrone, G., 2003. Distributed Intelligent Control of Exceptions in Reconfigurable Manufacturing Systems. International Journal Production Research, 41, 1393-1412.

Burbidge, J. L., 1989. Production Flow Analysis for Planning Group Technology. Clarendon Press. Oxford, England.

Buzacott, J., Shanthikumar, J. G., 1993. Stochastic Models of Manufacturing Systems, Prentice-Hall, Upper Saddle River, NJ, 103-106.

Catay, B., Vakharia, A., Erenguc, S., 2002. A Methodology for Component Grouping and Printed Circuit Board Scheduling in an Open Shop Manufacturing Environment, working paper. http://bear.cba.ufl.edu/vakharia/vita/openshop.doc June 15, 2006.

Caux, C., David, F. and Pierreval, H., 2006. Implementation of Delayed Differentiation in Batch Process Industries: A Standardization Problem, International Journal of Production Research, 44 (14), pp. 3243-3255.

Coronado, A. E., Lyons, A.G., Kehoe, D.F., Coleman, J., 2004. Enabling Mass Customization: Extending Build-to-Order concepts to Supply Chains. Production Planning and Control, 15 (4), 398-411.

De Lit, P., 2003. An Integrated Approach for Product Family and Assembly System Design. IEEE Transactions on Robotics and Automation, 19 (2), 324-334.

De Lit, P., L'Eglise, T., Delchambre, A., 2001. Functional Entities: A Concept to Support Product Family and Assembly System Design. Proceedings of the 4th IEEE, International Symposium on Assembly and Task Planning, Soft Research Park, Fukuoka, Japan. 160-165.

D'Souza, B, and Simpson, T. W., 2002. A Genetic Algorithm Based Method for Product Family Design Optimization. Engineering Optimization, 35 (1), 1-18.

Duray, R., Ward, P., Milligan, G., Berry, W., 2000. Approaches to Mass Customization: Configuration and Empirical Validation. Journal of Operation Management, 18 (6), 605-625.

Escoto, R. P., McDonnell L. R., Salort, E. V., Esteban, F. C. L., 1998. Development of an Algorithm for the Application of Group Technology in the Design of Manufacturing Systems. Production Planning & Control, 9 (3), 267-274.

Fan, I.-S. and Liu, C.-K., 1999. Product Family and Variants: Definition and Models, in Flexible Automation and Intelligent Manufacturing. J. Ashayeri, W. G. Sullivan, and M. M. Ahmad, Eds. New York: Begell House, 267-277.

Feitziner E. and Lee H. L., 1997, Mass Customisation at Hewlett-Packard: The Power of Postponement. Harvard Business Review, 75, 116–121.

Fung, R. Y. K., Chen, Y. and Tang, J., 2006. Estimating the Functional Relationships for Quality Function Deployment under Uncertainties. Fuzzy Sets and Systems, 157, 98-120.

Garg, A. and Tang, C. S., 1997. On Postponement Strategies for Product Families with Multiple Points of Differentiation. IIE Transactions, 29, 641-650.

Garud, R. and Kumaraswamy, A., 1995. Technological and Organizational Designs for Realizing Economies of Substitution. Strategic Management Journal, 16 (Special Issue), 93-109.

Golany, B., Dar-El, E. M. and Zeev, N., 1999. Controlling Shop Floor Operations in a Multi-family Multi-cell Manufacturing Environment through Constant Work-in-process. IIE Transactions, 31 (8), 771-781.

Gupta, S. and Krishnan, V. 1997. Product Family-Based Assembly Sequence Design Methodology. IIE Transactions, 30, 933-945.

Heikkila, A. and Koivo, H., 1998. Modular Control System with Intelligent Scheduling. Computers in Industry, 36 (1-2), 75-81.

Hermann, J.W. and Chincolkar, M. M., 2000. Design for Production: A Tool for Reducing Manufacturing Cycle Time. ASME Design Engineering Technical Conferences, ASME, Paper No. DFM-14002, Las Vegas, NV.

Jiao, J., Simpson, T.W. and Siddique, Z., 2007. Product Family Design and Platform-Based Product Development: A State-of-the-Art Review. Special Issue on Product Family Design and Platform-Based Product Development Journal of Intelligent Manufacturing. 18 (1), 5-29.

Jiao, J. and Tseng, M.M., 1999. A Methodology of Developing Product Family Architecture for Mass Customization. Journal of Intelligent Manufacturing 10 (1), 3-20.

Jiao, J. and Tseng, M. M., 2004. Customizability Analysis in Design for Mass Customization, Computer-Aided Design, 36, (8), 745-757.

Jiao, J., Zhang, L. and Pokharel, S., 2005. Coordinating Product and Process Variety for Mass Customized Order Fulfillment. Production Planning & Control, 16 (6), 608-620.

Kattan, I., Mikolajczak, B., Kattan, K. and Alqaassar, B., 2003. Minimizing Cycle Time and Group Scheduling using Petri Nets A Study of Heuristic Methods. Journal of Intelligent Manufacturing, 14 (1), 107-121.

Kariyakar, D. K. A., 2000. A Simulation-Based Approach for Assessing Delayed Product Differentiation Strategies for Assembling Families of Products. A Thesis in Industrial Engineering, The Pennsylvania State University, University Park, PA.

Krishnan, V. and Gupta, S., 2001, Appropriateness and Impact of Platform-based Product Development, Management Science, 47 (1), 52-68.

Kusiak, A., 2002. Integrated Product and Process Design: A Modularity Perspective. Journal Engineering Design, 13 (3), 223-231.

Kusiak, A. and He, D. W., 1998. Design for Agility: A Scheduling Perspective. Robotics Computer Integer Manufacturing, 14 (4), 415-27.

Lambert, S. Abdulnour, G. and Drolet, J., 2006. Flexibility Analysis of a Surface Mount Technology Electronic Assembly Plant: An Integrated Model Using Simulation. International Journal Flexibility Manufacturing System, 17 (2), 151-167.

Lee, H. L. and Tang, C. S., 1997. Modelling the Costs and Benefits of Delayed Product Differentiation. Management Science, 43 (1), 40-53.

Li, E. Y., Shaw, W., H. Jr. and Martin-Vega, L. A., 1996. Flow-time of Modified Scheduling Heuristics in a Dynamic Rescheduling Environment. Computers Industrial Engineering, 31 (1), 213-216.

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

Lourenco, L. L. and Pato, M. V., 2004. An Improved Genetic Heuristic to Support the Design of Flexible Manufacturing Systems. Computers & Industrial Engineering, 46 (1), 141-157.

Ma, S., Wang, W., and Liu, L., 2002. Commonality and Postponement in Multistage Assembly Systems. European Journal of Operations Research, 142 (3), 523-538.

Mangun, D. and Thurston, D. L., 2002. Incorporating Component Reuse, Remanufacture, and Recycle into Product Portfolio Design. IEEE Transactions on Engineering Management, 49 (4), 479-490.

McCutcheon, D. M., Raturi, A. S. and Meredith J. R., 1994. The Customization-Responsiveness Squeeze. Sloan Management Review, 35 (2), 89-99.

Meyer, M. and Lehnerd, A.P., 1997. The Power of Product Platform-Building Value and Cost Leadership. Free Press, New York.

Mikkola, J. H. and Gassman, O., 2003. Managing Modularity of Product Architectures: Toward and Integrated Theory. IEEE Transactions on Engineering Management, 50 (2), 204-218.

Mitrofanov. S. P., 1966. The Scientific Principles of Group Technology. National Lending Library Translation. 105-115.

Nagarur, N. and Azeem, A., 1999. Impact of Commonality and Flexibility on Manufacturing Performance: A Simulation Study. International Journal of Production Economics, 60-61 125-134.

Offodile, O. F., 1991. Application of Similarity Coefficient Method to Parts Coding Classification Analysis in Group Technology. Journal of Manufacturing Systems, 10 (1), 442-448.

Pahl, G. and Beitz, W., 1988. Engineering Design: A Systematic Approach, New York: Springer Verlag.

Park, J., 2005. Methods for Incorporating Cost Information into Platform-based Product Development. Department of Industrial and Manufacturing Engineering, Pennsylvania State University, PhD Dissertation.

Park, J. and Simpson, T. W., 2005. Development of a Production Cost Estimation Framework to Support Product Family Design. International Journal of Production Research, 42 (4), 731-772.

Pine, J., Victor, B. and Baynton A., 1993. Making Mass Customization Work. Harvard Business Review, 71 (5), 108-119.

Rhee, I., 1998. A Modular Algorithm for Resource Allocation. Distributed Computing, 11, (3), 157-168.

Saitou, K., Malpathak, S. and Qvam, H., 2002. Robust Design of Flexible Manufacturing Systems Using Colored Petri Net and Genetic Algorithm. Journal of Intelligent Manufacturing, 13 (5), 339-351.

Salum, L., 2000. The Cellular Manufacturing Layout Problem. International Journal Production Research, 38 (5), 1053-1069.

Salvador, F., Forza, C. and Rungtusanatham, M., 2002. Modularity, Product Variety, Production Volume, and Component Sourcing: Theorizing Beyond Generic Prescriptions. Journal of Operations Management, 20, 549-575.

Salvador, F. and Forza, C., 2004. Configuring Products to Address the Customization-Responsiveness Squeeze: A Survey of Management Issues and Opportunities. International Journal Production Economics, 91 (3), 273-291.

Sanchez, R., 1999. Modular Architectures in the Marketing Process. Journal of Marketing, 63 (Special Issue), 92-111.

Sawhney, M.S., 1998. Leveraged High-Variety Strategies: From Portfolio Thinking to Platform Thinking. Journal of Academy of Marketing Science, 26 (1), 54-61.

Schierholt, K., 2001. Press Configuration: Combining the Principles of Product Configuration and Process Planning. Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM, 15 (5):411-424.

Shirley, G.V., 1990. Models for Managing the Redesign and Manufacture of Product Sets. Journal of Manufacturing and Operations Management, 3 (2), 85-104.

Simpson, T.W., Maier, J.R.A. and Mistree, F., 2001. Product Platform Design: Method and Application. Research Engineering Design, 13(1), 2-22.

Simpson, T.W. and D'Souza, B., 2004. Assessing Variable Levels of Platform Commonality within a Product Family Using a Multiobjective Generic Algorithm. Concurrent Engineering: Research and Applications, 12(2), 119-130.

Skipworth, H. and Harrison, A., 2004 "Implications of Form Postponement to Manufacturing: A Case Study," International Journal of Production Research, 42 (10), 2063-2081.

Su, J. C.P., Chang, Y. and Ferguson, M., 2005. Evaluation of Postponement Structures to Accommodate Mass Customization. Journal of Operations Management, 23, 305-318.

Swaminathan, J. and Tayur, S. R., 1998. Managing Broader Product Lines through Delayed Differentiation Using Vanilla Boxes. Management Science, 44 (12), Part 2 of 2,161-172.

Swink, M., 2003. Completing Projects on-time: How Project Acceleration Affects New Product Development. Journal of Engineering Technology Management, 20 (4), 319-344.

Tauhid, S., Artar, H. U., Gupta, S. and Okudan, G., 2007. An Investigation of the Impact of Assembly Sequencing on the Product Family Design Outcomes, Proceedings of the ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference. DETC2007-35672.

Ulrich K. T., 1994. Fundamentals of Product Modularity. In S. Dasu (ed.), Management of Design: Engineering and Management Perspectives Boston: Kluwer, 219–229.

Ulrich, K., 1995. The Role of Product Architecture in the Manufacturing Firm, Research Policy, 24 (3), 419-440.

Van Der Zee, D. J., 2004. Dynamic Scheduling of Batch Servers with Compatible Product Families. International Production Research, 42 (22), 4803-4826.

Van Hoek, R. I., 2001. The Rediscovery of Postponement: A Literature Review and Directions For Research. Journal of Business Logistics, 19 (2), 161-184.

Zinn, W. and Bowersox, D.J., 1988. Planning Physical Distribution with the Principle of Postponement. Journal of Business Logistics, 9, (2), 117-136.

Appendix A

Bill of Materials (Park, 2005)

P2 Bill of Materials							
Subassembly	**Part Name**	**Quantity**	**Mass (gr.)**	**Material**	**Manuf. Process**	**Dimensions**	**Cost**
M1 Bit	Bit	1	13.7	M	Purchasing	L=63.5 D=6.3	$1.06
M2 Housing	Lower Housing	1	44.9	ABS	Injection	L=231	$0.90
	Upper Housing	1	41.4	ABS	Injection	L=231	-
	Screw	4	0.65	M	-	L=14.5 D=3	$0.05
M3 Lock	Lock	1	0	P->M	Purchasing	D=25.4	-
M4 Position	-	-	-	-	-	-	-
M5 Front End	-	-	-	-	-	-	-
M6 Gear	Ring Gear	1	63.5	P->M	Injection	L=25 D=30.5	-
	Planetary Arm	1	7.6	M	Powder Metallur.	L=20.2 T=4	$0.71
	Planetary	3	2.9	M	Purchasing	D=11.8 L=6.3	$0.58
	Intermediate	1	7.6	M	Powder Metallur.	L=20.2 T=4	$0.58
	Planetary	3	0.16	P	Purchasing	D=11.8 T=2	$0.82
	Cover	1	4	P->M	Stamping	H=10.8 D=30	$0.21
	Motor Coupling	1	1.1	M	Powder Metallur.	H=10.9	$0.04
M7 Shaft	Gear to Train	1	27.6	M	Powder Metallur.	L=12.7 D=4.8	$0.3
	Chuck	1	0	M	Powder Metallur.	L=5.3 D=13.4	-
	Bushing	1	2.2	M	-	D=16 L=6.3	-
	Shaft	1	-	M	-	-	$0.30
	Spring Washer	1	-	M	-	-	-
	Main Shaft	1	-	M	-	-	-
M8 Electric	Switch	1	14.9	P	Purchased	L=41.8	$1.07
	Switch Button	1	0	P	Purchased	L=25.5	-
	Battery (3.6 v)	1	142	-	Purchased	D=24.4	$5.97
	Battery Cap	1	-	-	Purchased	-	-
	Motor	1	64	-	Purchased	D=27.5 L=51	$1.13
M9 Head	-	-	-	-	-	-	-

P3 Bill of Materials							
Subassembly	**Part Name**	**Quantity**	**Mass (gr.)**	**Material**	**Manuf. Process**	**Dimensions**	**Cost**
M1 Bit	Bit	1	13.7	M	Purchasing	L=63.5 D=6.3	$1.06
	Bit Storage	1	-	P	Purchasing	L=73.4	-
M2 Housing	Lower Housing	1	44.9	ABS	Injection	L=244.6	$0.90
	Upper Housing	1	41.4	ABS	Injection	L=244.6	-
	Screw	4	0.65	M	-	L=14.5 D=3	$0.05
M3 Lock	Outside Torque	1	19.6	M	Powder Metallur.	H=7.6	-
	Roller	2	0.6	M	Purchasing	H=6.3 D=4	-
M4 Position	-	-	-	-	-	-	-
M5 Front End	-	-	-	-	-	-	-
M6 Gear	Ring Gear	1	63.5	P->M	Injection	L=25 D=30	-
	Planetary Arm	1	7.6	M	Powder Metallur.	L=20.2 T=4	$0.71
	Planetary	3	2.9	M	Purchasing	D=11.8 L=6.3	$0.58
	Intermediate	1	7.6	M	Powder Metallur.	L=20.2 T=4	$0.58
	Planetary	3	0.16	P	Purchasing	D=11.8 T=2	$0.82
	Cover	1	4	P->M	Stamping	H=10.8 D=30	-
	Motor Coupling	1	1.1	M	Powder Metallur.	H=10.9	$0.04
M7 Shaft	Gear to Train	1	27.6	M	Powder Metallur.	L=11.6 D=6	$0.30
	Chuck	1	0	M	Powder Metallur.	L=5.3 D=13.4	$0.83
	Bushing	1	2.2	M	-	D=16 L=6.3	-
	Shaft	1	-	M	-	-	$0.30
	Spring Washer	1	-	M	-	-	-
	Main Shaft	1	-	M	-	-	-
M8 Electric	Switch	1	14.9	P	Purchased	L=41.8	$1.07
	Switch Button	1	0	P	Purchased	L=25.5	-
	Battery (3.6 v)	1	142	-	Purchased	D=24.4	$5.97
	Battery Cap	1	-	-	Purchased	D=30 L=20	-
	Motor	1	64	-	Purchased	D=27.5 L=51	$1.13
M9 Head	-	-	-	-	-	-	-

P4 Bill of Materials

Subassembly	Part Name	Quantity	Mass (gr.)	Material	Manuf. Process	Dimensions	Cost
M1 Bit	Bit	2	13.7	M	Purchasing	L=63.5 D=6.3	$2.12
M2 Housing	Lower Housing	1	32	ABS	Injection	L=148	$2.25
	Upper Housing	1	41.4	ABS	Injection	L=148	$0.25
M3 Lock	Outside Torque	1	11.5	M	Powder Metallur.	H=6	-
	Inside Torque	1	2.9	M	Powder Metallur.	L=15.5	-
	Roller	3	0.4	M	Purchasing	-	-
M4 Position	Lock Handle	1	1.2	P	-	L=28	-
	Wedge	1	1	P	-	L=25	-
	Spring	1	0.3	M	Purchased	D=4.9	-
	Position Gear	1	1.5	P	Injection	L=19.2	-
M5 Front End	Lower Housing	1	15.6	ABS	Injection	L=71	-
	Upper Housing	1	15.1	ABS	Injection	L=71	-
	Snap Ring	1	5.1	M	-	2 leg =25	-
M6 Gear	Gear Housing	1	83.6	M	Die-Casting	L=62	$1.22
	Housing Pin	2	-	M	-	L=40	-
	Ring Gear	1	5.8	P->M	Injection	L=20	$0.59
	Planetary Arm	1	20.2	M	Powder Metallur.	L=44	$0.61
	Planetary	3	4	M	Purchasing	D=14.6	$0.88
	Intermediate	1	14	M	Powder Metallur.	L=25	$0.60
	Planetary	3	0.6	P	Purchasing	D=14.6	$0.83
	Cover	1	4	P->M	Stamping	D=37	-
	Motor Coupling	1	-	M	Powder Metallur.	D=5.6	$0.02
M7 Shaft	Gear to Train	1	-	M	Powder Metallur.	D=6.9	$0.30
	Chuck	1	0	M	Powder Metallur.	D=10	$0.87
M8 Electric	Switch	1	-	P	Purchasing	-	$1.07
	Switch Button	1	-	P	Purchasing	L=32	-
	Battery (2.4 v)	2	35	-	Purchasing	-	$5.55
	Motor	1	77	-	Purchasing	L=51	$1.13

M9 Head	-	-	-	-	-	-	-
	Light	2	0	-	Purchasing	D=44.3	$0.1
	Head Housing	1	11.5	P	Injection	D=44.3	-
	Head Middle	1	3	P	Injection	D=44.7	-
	Head Cap	1	4.6	P	Injection	D=41	-
	Electric	4	2	M	Purchasing	D=40	-
	Snap Ring	1	0.6	M	Purchasing	D=20	-
	Washer	1	-	M	Purchasing	D=20	-

P5 Bill of Materials

Subassembly	Part Name	Quantity	Mass (gr.)	Material	Manuf. Process	Dimensions	Cost
M1 Bit	Bit	2	13.7	M	Purchasing	L=63.5 D=6.3	$2.12
M2 Housing	Lower Housing	1	28.8	ABS	Injection	L=172	$2.25
	Upper Housing	1	32	ABS	Injection	L=172	$0.25
	Snap Ring	2	0.7	M	-	D=20	-
	Side Cap	2	0.5	P	Injection	D=9.9	-
	Hinge Ring	2	0.8	M	Die-Casting	D=9.9	-
	Long Screw	1	1.6	M	-	D=2.85	-
	Hand Cover	2	-	TPE	-	T=2	-
M3 Lock	Outside Torque	1	11.5	M	Powder Metallur.	H=6	-
	Inside Torque	1	2.9	M	Powder Metallur.	L=15.5	-
	Roller	3	0.4	M	Purchasing	-	-
M4 Position	Lock Handle	1	1.2	P	-	L=28	-
	Wedge	1	1	P	-	L=25	-
	Spring	1	0.3	M	Purchased	D=4.9	$0.05/kg
	Position Gear	1	1.5	P	Injection	L=19.2	-
M5 Front End	Lower Housing	1	15.6	ABS	Injection	L=71	-
	Upper Housing	1	15.1	ABS	Injection	L=71	-
	Snap Ring	1	5.1	M	-	2 leg =25	-
M6 Gear	Gear Housing	1	83.6	M	Die-Casting	L=62	$1.22
	Housing Pin	2	-	M	-	L=40	-
	Ring Gear	1	5.8	P->M	Injection	L=20	$0.59
	Planetary Arm	1	20.2	M	Powder Metallur.	L=44	$0.61
	Planetary	3	4	M	Purchasing	D=14.6	$0.88
	Intermediate	1	14	M	Powder Metallur.	L=25	$0.60
	Planetary	3	0.6	P	Purchasing	D=14.6	$0.83
	Cover	1	4	P->M	Stamping	D=37	-
	Motor Coupling	1	-	M	Powder Metallur.	D=5.6	$0.02
M7 Shaft	Gear to Train	1	-	M	Powder Metallur.	D=6.9	$0.30
	Chuck	1	0	M	Powder Metallur.	D=10	$0.87
M8 Electric	Switch	1	-	P	Purchasing	-	$1.07
	Switch Button	1	-	P	Purchasing	L=32	-
	Connector	1	1.3	M	Purchasing	-	-

	Battery (3.6 v)	1	134.8	-	Purchasing	-	$7.25
	Motor	1	77	-	Purchasing	L=51	$1.13
	Wire	4	-	Wire	Purchasing	L=10	$0.1/m
M9 Head	-	-	-	-	-	-	-
	Light	2	0	-	Purchasing	D=44.3	$0.1
	Head Housing	1	11.5	P	Injection	D=44.3	-
	Head Middle	1	3	P	Injection	D=44.7	-
	Head Cap	1	4.6	P	Injection	D=41	-
	Electric	4	2	M	Purchasing	D=40	-
	Snap Ring	1	0.6	M	Purchasing	D=20	-
	Washer	1	-	M	Purchasing	D=20	-

L=Length, DM=Max Diameter, Dm=Min Diameter, T=Thickness, W=Width, H= Height

Appendix B

Siman Codes for the 100 K Demand

```
PROJECT,      "Unnamed Project","Hakan Artar",,,No,Yes,Yes,Yes,No,No,No,No,No,No;
ATTRIBUTES:   component:
              keeptrack product:
              demand1:
              demand2:
              order time10:
              demand3:
              order time11:
              demand4:
              order time12:
              demand5:
              order time13:
              order time14:
              order time15:
              part type:
              order content:
              starttime1:
              starttime2:
              starttime3:
              starttime4:
              starttime5:
              order time1:
              order time2:
              order time3:
              order time4:
              order time5:
              order time6:
              order time7:
              order time8:
              order time9:
              process
time(5,42),1,1,1,1,1,170,170,1,170,170,160,160,170,160,160,1,1,160,0,1,1,1,1,0,10,20,20,0,1,250,0,0,1,1,1,
1,1,20,1,1,0,0,0,1,0,1,1,1,42,0,0,1,0,44,1,0,0,1,44,1,0,0,1,1,1,1,0,0,1070,1,1,1,0,0,42,1,1,0,20,44,1,1,1,0,44,0,
1,1,1,1,0,1,1,0,1070,0,1,1,1,0,0,1,1,1,20,0,1,1,0,0,0,1,1,0,1,0,0,1,0,0,0,0,1,1,1,0,0,0,1,1,0,0,0,1,0,0,0,0,1,0,0,
0,0,1,0,0,0,0,71,1,0,0,0,30,1,0,0,0,26,1,0,0,0,1,1,0,0,0,1,1,0,0,0,1,1,0,0,0,0,1,0,0,0,0,71,0,0,0,0,30,0,0,0,0,26,
0,0,0,0,1,0,0,0,0,1,0,0,0,0,1;
VARIABLES:    p3 out of order content p2 and p3.NumberOut
Dup,CLEAR(Statistics),CATEGORY("Exclude"):
              counter 0?.NumberOut False,CLEAR(Statistics),CATEGORY("Exclude"):
              p5 out of order content p3 and p5.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
              assembly line5.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
              assemble part3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              p3 out of order content p1 and p3.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
              Dispose 8.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
```

```
delay15.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
create components5.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
p2 out of order content p1 and p2.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
delay11.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
p3 out of order content p3 and p4.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
powder metallurgy process.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
assemble order content6.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
p1 drivers.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
delay10.NumberIn,CLEAR (Statistics),CATEGORY("Exclude"):
assembly line1.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
assemble order content9.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
p3 drivers.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
create screwdriver.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
delay12.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
delay14.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
delay5.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
delay8.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
assembly line4.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
delay12.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
purchase bit.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
create components.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
p1 out of order content p1 and p3.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
duplicatedproducts,CLEAR(System),CATEGORY("User Specified-User Specified"):
assembly line1.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
purchase bit.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
p1 out of order content p1 and p4.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
assemble order content3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
assembly line3.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
p4 out of order content p4 and p5.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
p4 out of order content p2 and p4.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
p4 drivers.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
counter 0?2.NumberOut False,CLEAR(Statistics),CATEGORY("Exclude"):
assembly line5.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
delay2.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
delay1.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
assembly line2.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
delay3.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
delay5.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
delay5.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
delay7.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
delay9.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
p3 out of order content p3 and p5.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
p2 out of order content p2 and p5.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
counter 0?4.NumberOut True,CLEAR(Statistics),CATEGORY("Exclude"):
powder metallurgy process.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
create components3.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
delay13.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
diecasting process.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
p5 out of order content p1 and p5.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
assemble order content12.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
```

p2 out of order content p1 and p2.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
p5 out of order content p4 and p5.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
assemble order content15.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
diecasting process.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
p4 drivers.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
assemble part5.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
delay7.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
p5 out of order content p3 and p5.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
delay2.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
p5 out of order content p4 and p5.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
counter 0?.NumberOut True,CLEAR(Statistics),CATEGORY("Exclude"):
p2 out of order content p2 and p3.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
p3 out of order content p3 and p4.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
assembly line4.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
assemble order content8.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
assemble part2.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
assembly line3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
delay11.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
p1 out of order content p1 and p4.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
delay15.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
p1 out of order content p1 and p5.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
delay14.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
p2 drivers.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
delay10.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
counter 0?3.NumberOut False,CLEAR(Statistics),CATEGORY("Exclude"):
injection process.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
assemble order content2.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
cdemand1,CLEAR(System),CATEGORY("User Specified-User Specified"):
cdemand2,CLEAR(System),CATEGORY("User Specified-User Specified"):
delay1.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
cdemand3,CLEAR(System),CATEGORY("User Specified-User Specified"):
p5 out of order content p2 and p5.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
cdemand4,CLEAR(System),CATEGORY("User Specified-User Specified"):
assemble order content5.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
cdemand5,CLEAR(System),CATEGORY("User Specified-User Specified"):
p3 out of order content p1 and p3.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
delay9.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
delay4.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
p5 drivers.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
counter 0?5.NumberOut True,CLEAR(Statistics),CATEGORY("Exclude"):
delay7.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
delay4.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
create components3.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
p5 out of order content p1 and p5.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
p4 out of order content p3 and p4.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
p5 drivers.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
assemble order content14.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
p4 out of order content p3 and p4.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
say,CLEAR(System),CATEGORY("User Specified-User Specified"):
injection process.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):

```
assembly line1.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
delay11.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
delay13.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
delay15.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
p5 out of order content p2 and p5.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
p2 out of order content p2 and p4.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
create components4.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
counter 0?4.NumberOut False,CLEAR(Statistics),CATEGORY("Exclude"):
injection process.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
delay12.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
p3 out of order content p3 and p5.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
assemble order content11.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
p4 out of order content p4 and p5.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
assembly line2.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
assembly line4.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
assemble part1.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
p2 out of order content p2 and p3.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
counter 0?2.NumberOut True,CLEAR(Statistics),CATEGORY("Exclude"):
diecasting process.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
delay2.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
assembly line2.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
assemble part4.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
delay4.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
p1 out of order content p1 and p5.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
delay6.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
delay6.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
delay8.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
assembly line5.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
p4 out of order content p2 and p4.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
delay13.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
delay1.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
p3 out of order content p2 and p3.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
counter(400),CLEAR(System),CATEGORY("User Specified-User Specified"),0:
assemble order content1.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
p1 out of order content p1 and p2.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
create components part2.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
p2 out of order content p2 and p5.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
assemble order content4.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
p4 out of order content p1 and p4.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
create components.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
create components4.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
assembly line3.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
assemble order content7.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
delay6.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
p4 out of order content p1 and p4.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
p1 drivers.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
p3 drivers.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
delay10.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
delay9.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
create components part2.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
```

```
            delay14.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
            powder metallurgy process.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
            purchase bit.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
            p1 out of order content p1 and p2.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
            delay8.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
            delay3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
            create components5.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
            p2 drivers.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
            p1 out of order content p1 and p3.NumberOut Dup,CLEAR(Statistics),CATEGORY("Exclude"):
            counter 0?5.NumberOut False,CLEAR(Statistics),CATEGORY("Exclude"):
            delay3.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"):
            assemble order content10.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
            p2 out of order content p2 and p4.NumberOut Orig,CLEAR(Statistics),CATEGORY("Exclude"):
            counter 0?3.NumberOut True,CLEAR(Statistics),CATEGORY("Exclude"):
            assemble order content13.NumberOut,CLEAR(Statistics),CATEGORY("Exclude");
QUEUES:     assemble order content1.Queue,FIFO,,AUTOSTATS(Yes,,):
            assemble part3.Queue,FIFO,,AUTOSTATS(Yes,,):
            assemble order content15.Queue,FIFO,,AUTOSTATS(Yes,,):
            assemble order content7.Queue,FIFO,,AUTOSTATS(Yes,,):
            assemble order content10.Queue,FIFO,,AUTOSTATS(Yes,,):
            assemble order content2.Queue,FIFO,,AUTOSTATS(Yes,,):
            assemble part4.Queue,FIFO,,AUTOSTATS(Yes,,):
            purchase bit.Queue,FIFO,,AUTOSTATS(Yes,,):
            injection process.Queue,FIFO,,AUTOSTATS(Yes,,):
            assemble order content11.Queue,FIFO,,AUTOSTATS(Yes,,):
            assemble order content3.Queue,FIFO,,AUTOSTATS(Yes,,):
            assemble part5.Queue,FIFO,,AUTOSTATS(Yes,,):
            assemble order content8.Queue,FIFO,,AUTOSTATS(Yes,,):
            assemble order content12.Queue,FIFO,,AUTOSTATS(Yes,,):
            diecasting process.Queue,FIFO,,AUTOSTATS(Yes,,):
            assemble order content4.Queue,FIFO,,AUTOSTATS(Yes,,):
            assemble order content9.Queue,FIFO,,AUTOSTATS(Yes,,):
            assemble part1.Queue,FIFO,,AUTOSTATS(Yes,,):
            assemble order content13.Queue,FIFO,,AUTOSTATS(Yes,,):
            assemble part2.Queue,FIFO,,AUTOSTATS(Yes,,):
            assemble order content5.Queue,FIFO,,AUTOSTATS(Yes,,):
            powder metallurgy process.Queue,FIFO,,AUTOSTATS(Yes,,):
            assemble order content14.Queue,FIFO,,AUTOSTATS(Yes,,):
            assemble order content6.Queue,FIFO,,AUTOSTATS(Yes,,);
PICTURES:   Picture.Airplane:
            Picture.Green Ball:
            Picture.Blue Page:
            Picture.Telephone:
            Picture.Blue Ball:
            Picture.Yellow Page:
            Picture.EMail:
            Picture.Yellow Ball:
            Picture.Bike:
            Picture.Report:
            Picture.Van:
```

```
            Picture.Widgets:
            Picture.Envelope:
            Picture.Fax:
            Picture.Truck:
            Picture.Letter:
            Picture.Box:
            Picture.Woman:
            Picture.Package:
            Picture.Man:
            Picture.Diskette:
            Picture.Boat:
            Picture.Red Page:
            Picture.Green Page:
            Picture.Red Ball;
RESOURCES:    injection,Capacity(8),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes,,):
            powder
metallurgy,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes,,):
            diecasting,Capacity(10),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes,,):
            assembly
workers,Capacity(15),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes,,):
            purchase,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes,,);
COUNTERS:     countparttype2done,,,,DATABASE(,"Count","User Specified","countparttype2done"):
            countparttype4done,,,,DATABASE(,"Count","User Specified","countparttype4done"):
            countparttype1done,,,,DATABASE(,"Count","User Specified","countparttype1done"):
            countparttype3done,,,,DATABASE(,"Count","User Specified","countparttype3done"):
            countparttype5done,,,,DATABASE(,"Count","User Specified","countparttype5done");

TALLIES:      leadtime10 order,,DATABASE(,"Interval","User Specified","leadtime10 order"):
            leadtime1,,DATABASE(,"Interval","User Specified","leadtime1"):
            leadtime2,,DATABASE(,"Interval","User Specified","leadtime2"):
            leadtime3,,DATABASE(,"Interval","User Specified","leadtime3"):
            leadtime4,,DATABASE(,"Interval","User Specified","leadtime4"):
            leadtime5,,DATABASE(,"Interval","User Specified","leadtime5"):
            leadtime6 order,,DATABASE(,"Interval","User Specified","leadtime6 order"):
            leadtime11 order,,DATABASE(,"Interval","User Specified","leadtime11 order"):
            leadtime1 order,,DATABASE(,"Interval","User Specified","leadtime1 order"):
            leadtime2 order,,DATABASE(,"Interval","User Specified","leadtime2 order"):
            leadtime7 order,,DATABASE(,"Interval","User Specified","leadtime7 order"):
            leadtime12 order,,DATABASE(,"Interval","User Specified","leadtime12 order"):
            leadtime3 order,,DATABASE(,"Interval","User Specified","leadtime3 order"):
            leadtime8 order,,DATABASE(,"Interval","User Specified","leadtime8 order"):
            leadtime13 order,,DATABASE(,"Interval","User Specified","leadtime13 order"):
            leadtime9 order,,DATABASE(,"Interval","User Specified","leadtime9 order"):
            leadtime14 order,,DATABASE(,"Interval","User Specified","leadtime14 order"):
            leadtime4 order,,DATABASE(,"Interval","User Specified","leadtime4 order"):
            leadtime15 order,,DATABASE(,"Interval","User Specified","leadtime15 order"):
            leadtime5 order,,DATABASE(,"Interval","User Specified","leadtime5 order");
REPLICATE,    1,,HoursToBaseTime(500),Yes,Yes,,,,24,Hours,No,No,,,Yes;
ENTITIES:     screwdriver,Picture.Report,0.0,0.0,0.0,0.0,0.0,0.0,AUTOSTATS(Yes,,);
;
```

```
;    Model statements for module:  Create 2
;
155$        CREATE,       1,HoursToBaseTime(0.0),screwdriver:HoursToBaseTime(NORM(8 ,
2)):NEXT(156$);

156$        ASSIGN:       create screwdriver.NumberOut=create screwdriver.NumberOut + 1:NEXT(154$);
;
;    Model statements for module:  Assign 84
;
154$        ASSIGN:       order content=
DISC(.066,1,.132,2,.198,3,.264,4,.33,5,.396,6,.462,7,.528,8,.594,9,.66,10,.726,11,.792,12,.858,13,.924,14,1,
15):
                    duplicatedproducts=duplicatedproducts + 1:
                    keeptrack product=duplicatedproducts:NEXT(47$);
;
;    Model statements for module:  Decide 19
;
47$         BRANCH,       1:
                    If,order content==1,48$,Yes:
                    If,order content==2,50$,Yes:
                    If,order content==3,52$,Yes:
                    If,order content==4,54$,Yes:
                    If,order content==5,56$,Yes:
                    If,order content==6,58$,Yes:
                    If,order content==7,62$,Yes:
                    If,order content==8,66$,Yes:
                    If,order content==9,70$,Yes:
                    If,order content==10,74$,Yes:
                    If,order content==11,78$,Yes:
                    If,order content==12,82$,Yes:
                    If,order content==13,86$,Yes:
                    If,order content==14,90$,Yes:
                    Else,94$,Yes;
;
;    Model statements for module:  Separate 34
;
94$         DUPLICATE,    100 - 50:
                    NORM(166,16),163$,50:NEXT(162$);
162$        ASSIGN:       p5 out of order content p4 and p5.NumberOut Orig=
                    p5 out of order content p4 and p5.NumberOut Orig + 1:NEXT(95$);
163$        ASSIGN:       p5 out of order content p4 and p5.NumberOut Dup=p5 out of order content p4 and
p5.NumberOut Dup + 1
                    :NEXT(97$);
;
;    Model statements for module:  Separate 35
;
95$         DUPLICATE,    100 - 50:
                    NORM(166,16),166$,50:NEXT(165$);
165$        ASSIGN:       p4 out of order content p4 and p5.NumberOut Orig=
                    p4 out of order content p4 and p5.NumberOut Orig + 1:NEXT(96$);
```

```
166$        ASSIGN:       p4 out of order content p4 and p5.NumberOut Dup=p4 out of order content p4 and
p5.NumberOut Dup + 1
                         :NEXT(96$);

;
;    Model statements for module:  Assign 81
;
96$         ASSIGN:       order time15=TNOW:
                         part type=4:
                         counter(keeptrack product)=counter(keeptrack product) + 1:NEXT(0$);
;
;    Model statements for module:  Decide 2
;
0$          BRANCH,       1:
                         If,part type==1,25$,Yes:
                         If,part type==2,30$,Yes:
                         If,part type==3,35$,Yes:
                         If,part type==4,40$,Yes:
                         Else,45$,Yes;
;
;    Model statements for module:  Assign 42
;
45$         ASSIGN:       cdemand5=cdemand5+1:
                         demand5=cdemand5:
                         starttime5=TNOW:NEXT(41$);;
;
;    Model statements for module:  Separate 10
;
41$         DUPLICATE,    100 - 0:
                         41,171$,0:NEXT(170$);

170$        ASSIGN:       create components5.NumberOut Orig=create components5.NumberOut Orig +
1:NEXT(42$);
171$        ASSIGN:       create components5.NumberOut Dup=create components5.NumberOut Dup +
1:NEXT(42$);
;
;    Model statements for module:  Assign 40
;
42$         ASSIGN:       say=say + 1:
                         component=say:NEXT(43$);
;
;    Model statements for module:  Decide 18
;
43$         BRANCH,       1:
                         If,component==42,172$,Yes:
                         Else,173$,Yes;
172$        ASSIGN:       counter 0?5.NumberOut True=counter 0?5.NumberOut True + 1:NEXT(44$);
173$        ASSIGN:       counter 0?5.NumberOut False=counter 0?5.NumberOut False + 1:NEXT(5$);
;
;    Model statements for module:  Assign 41
```

```
;
44$          ASSIGN:       say=0:NEXT(5$);
;
;     Model statements for module:  Decide 12
;
5$          BRANCH,       1:
                          If,
                          component == 1 .or. component == 4 .or. component == 7 .or. component == 8 .or.
component == 11 .or. component == 12 .or. component == 13 .or. component == 14 .or. component == 18
.or. component == 20 .or. component == 23 .or. component == 25 .or. component == 26 .or. component ==
30 .or. component == 31 .or. component == 32 .or. component == 33 .or. component == 34 .or. component
== 35 .or. component == 36 .or. component == 40 .or. component == 41 .or. component == 42,
                          6$,Yes:
                          If,
                          component == 2 .or. component == 3 .or. component == 5 .or. component == 15 .or.
component == 16 .or. component == 17 .or. component == 21 .or. component == 37 .or. component == 38
.or.component == 39,
                          7$,Yes:
                          If,component == 6 .or. component == 19,9$,Yes:
                          Else,8$,Yes;
;
;     Model statements for module:  Process 3
;
8$          ASSIGN:       powder metallurgy process.NumberIn=powder metallurgy process.NumberIn + 1:
                          powder metallurgy process.WIP=powder metallurgy process.WIP+1;
179$        QUEUE,        powder metallurgy process.Queue;
178$        SEIZE,        2,VA:
                          powder metallurgy,1:NEXT(177$);
177$        DELAY:        0.000277777777778,,VA;
176$        RELEASE:      powder metallurgy,1;
224$        ASSIGN:       powder metallurgy process.NumberOut=powder metallurgy process.NumberOut +
1:
                          powder metallurgy process.WIP=powder metallurgy process.WIP-1:NEXT(10$);
;
;     Model statements for module:  Decide 13
;
10$         BRANCH,       1:
                          If,part type==1,11$,Yes:
                          If,part type==2,12$,Yes:
                          If,part type==3,13$,Yes:
                          If,part type==4,14$,Yes:
                          Else,15$,Yes;
;
;     Model statements for module:  Batch 5
;
15$         QUEUE,        assemble part5.Queue;
229$        GROUP,        demand5,Permanent:42,Last:NEXT(230$);
230$        ASSIGN:       assemble part5.NumberOut=assemble part5.NumberOut + 1:NEXT(106$);

;
```

```
;    Model statements for module:  Process 9
;
106$        ASSIGN:       assembly line5.NumberIn=assembly line5.NumberIn + 1:
                    assembly line5.WIP=assembly line5.WIP+1;
232$        DELAY:        SecondsToBaseTime(NORM(114,2)),,VA;
279$        ASSIGN:       assembly line5.NumberOut=assembly line5.NumberOut + 1:
                    assembly line5.WIP=assembly line5.WIP-1:NEXT(20$);
;
;    Model statements for module:  Record 5
;
20$         COUNT:        countparttype5done,1:NEXT(101$);
;
;    Model statements for module:  Record 10
;
101$        TALLY:        leadtime5,INT(starttime5),1:NEXT(107$);
;
;    Model statements for module:  Decide 21
;
107$        BRANCH,       1:
                    If,order content==1,139$,Yes:
                    If,order content==2,140$,Yes:
                    If,order content==3,141$,Yes:
                    If,order content==4,142$,Yes:
                    If,order content==5,143$,Yes:
                    If,order content==6,144$,Yes:
                    If,order content==7,145$,Yes:
                    If,order content==8,146$,Yes:
                    If,order content==9,147$,Yes:
                    If,order content==10,148$,Yes:
                    If,order content==11,149$,Yes:
                    If,order content==12,150$,Yes:
                    If,order content==13,151$,Yes:
                    If,order content==14,152$,Yes:
                    Else,153$,Yes;
;
;    Model statements for module:  Process 39
;
153$        ASSIGN:       delay15.NumberIn=delay15.NumberIn + 1:
                    delay15.WIP=delay15.WIP+1;
285$        DELAY:        0.000000027777778,,VA;
332$        ASSIGN:       delay15.NumberOut=delay15.NumberOut + 1:
                    delay15.WIP=delay15.WIP-1:NEXT(138$);
;
;    Model statements for module:  Batch 50
;
138$        QUEUE,        assemble order content15.Queue;
335$        GROUP,        keeptrack product,Permanent:counter(keeptrack product),Last:NEXT(336$);

336$        ASSIGN:       assemble order content15.NumberOut=assemble order content15.NumberOut +
1:NEXT(137$);
```

```
;
;    Model statements for module:  Record 56
;
137$         TALLY:         leadtime15 order,INT(order time15),1:NEXT(110$);
;
;    Model statements for module:  Dispose 8
;
110$         ASSIGN:        Dispose 8.NumberOut=Dispose 8.NumberOut + 1;
337$         DISPOSE:        Yes;
;
;    Model statements for module:  Process 25
;
139$         ASSIGN:        delay1.NumberIn=delay1.NumberIn + 1:
                      delay1.WIP=delay1.WIP+1;
339$         DELAY:         0.000000027777778,,VA;
386$         ASSIGN:        delay1.NumberOut=delay1.NumberOut + 1:
                      delay1.WIP=delay1.WIP-1:NEXT(109$);
;
;    Model statements for module:  Batch 36
;
109$         QUEUE,         assemble order content1.Queue;
389$         GROUP,         keeptrack product,Permanent:counter(keeptrack product),Last:NEXT(390$);

390$         ASSIGN:        assemble order content1.NumberOut=assemble order content1.NumberOut +
1:NEXT(108$);
;
;    Model statements for module:  Record 42
;
108$         TALLY:         leadtime1 order,INT(order time1),1:NEXT(110$);
;
;    Model statements for module:  Process 26
;
140$         ASSIGN:        delay2.NumberIn=delay2.NumberIn + 1:
                      delay2.WIP=delay2.WIP+1;
392$         DELAY:         0.000000027777778,,VA;
439$         ASSIGN:        delay2.NumberOut=delay2.NumberOut + 1:
                      delay2.WIP=delay2.WIP-1:NEXT(112$);
;
;    Model statements for module:  Batch 37
;
112$         QUEUE,         assemble order content2.Queue;
442$         GROUP,         keeptrack product,Permanent:counter(keeptrack product),Last:NEXT(443$);
443$         ASSIGN:        assemble order content2.NumberOut=assemble order content2.NumberOut +
1:NEXT(111$);


;
;    Model statements for module:  Record 43
;
111$         TALLY:         leadtime2 order,INT(order time2),1:NEXT(110$);
```

```
;
;    Model statements for module:  Process 27
;
141$        ASSIGN:       delay3.NumberIn=delay3.NumberIn + 1:
                          delay3.WIP=delay3.WIP+1;
445$        DELAY:        0.000000027777778,,VA;
492$        ASSIGN:       delay3.NumberOut=delay3.NumberOut + 1:
                          delay3.WIP=delay3.WIP-1:NEXT(114$);
;
;    Model statements for module:  Batch 38
;
114$        QUEUE,        assemble order content3.Queue;
495$        GROUP,        keeptrack product,Permanent:counter(keeptrack product),Last:NEXT(496$);
496$        ASSIGN:       assemble order content3.NumberOut=assemble order content3.NumberOut +
1:NEXT(113$);
;
;    Model statements for module:  Record 44
;
113$        TALLY:        leadtime3 order,INT(order time3),1:NEXT(110$);
;
;
;    Model statements for module:  Process 28
;
142$        ASSIGN:       delay4.NumberIn=delay4.NumberIn + 1:
                          delay4.WIP=delay4.WIP+1;
498$        DELAY:        0.000000027777778,,VA;
545$        ASSIGN:       delay4.NumberOut=delay4.NumberOut + 1:
                          delay4.WIP=delay4.WIP-1:NEXT(116$);
;
;    Model statements for module:  Batch 39
;
116$        QUEUE,        assemble order content4.Queue;
548$        GROUP,        keeptrack product,Permanent:counter(keeptrack product),Last:NEXT(549$);
549$        ASSIGN:       assemble order content4.NumberOut=assemble order content4.NumberOut +
1:NEXT(115$);
;
;    Model statements for module:  Record 45
;
115$        TALLY:        leadtime4 order,INT(order time4),1:NEXT(110$);
;
;    Model statements for module:  Process 29
;
143$        ASSIGN:       delay5.NumberIn=delay5.NumberIn + 1:
                          delay5.WIP=delay5.WIP+1;
551$        DELAY:        0.000000027777778,,VA;
598$        ASSIGN:       delay5.NumberOut=delay5.NumberOut + 1:
                          delay5.WIP=delay5.WIP-1:NEXT(118$);
;
;    Model statements for module:  Batch 40
;
```

```
118$        QUEUE,       assemble order content5.Queue;
601$        GROUP,       keeptrack product,Permanent:counter(keeptrack product),Last:NEXT(602$);
602$        ASSIGN:      assemble order content5.NumberOut=assemble order content5.NumberOut +
1:NEXT(117$);
;
;    Model statements for module:  Record 46
;
117$        TALLY:       leadtime5 order,INT(order time5),1:NEXT(110$);
;
;    Model statements for module:  Process 30
;
144$        ASSIGN:      delay6.NumberIn=delay6.NumberIn + 1:
                  delay6.WIP=delay6.WIP+1;
604$        DELAY:       0.000000027777778,,VA;
651$        ASSIGN:      delay6.NumberOut=delay6.NumberOut + 1:
                  delay6.WIP=delay6.WIP-1:NEXT(120$);


;
;    Model statements for module:  Batch 41
;
120$        QUEUE,       assemble order content6.Queue;
654$        GROUP,       keeptrack product,Permanent:counter(keeptrack product),Last:NEXT(655$);
655$        ASSIGN:      assemble order content6.NumberOut=assemble order content6.NumberOut +
1:NEXT(119$);
;
;    Model statements for module:  Record 47
;
119$        TALLY:       leadtime6 order,INT(order time6),1:NEXT(110$);
;
;    Model statements for module:  Process 31
;
145$        ASSIGN:      delay7.NumberIn=delay7.NumberIn + 1:
                  delay7.WIP=delay7.WIP+1;
657$        DELAY:       0.000000027777778,,VA;
704$        ASSIGN:      delay7.NumberOut=delay7.NumberOut + 1:
                  delay7.WIP=delay7.WIP-1:NEXT(122$);
;
;    Model statements for module:  Batch 42
;
122$        QUEUE,       assemble order content7.Queue;
707$        GROUP,       keeptrack product,Permanent:counter(keeptrack product),Last:NEXT(708$);
708$        ASSIGN:      assemble order content7.NumberOut=assemble order content7.NumberOut +
1:NEXT(121$);
;
;    Model statements for module:  Record 48
;
121$        TALLY:       leadtime7 order,INT(order time7),1:NEXT(110$);
;
;    Model statements for module:  Process 32
```

```
;
146$        ASSIGN:       delay8.NumberIn=delay8.NumberIn + 1:
                     delay8.WIP=delay8.WIP+1;
710$        DELAY:        0.000000027777778,,VA;
757$        ASSIGN:       delay8.NumberOut=delay8.NumberOut + 1:
                     delay8.WIP=delay8.WIP-1:NEXT(124$);
;
;    Model statements for module:  Batch 43
;
124$        QUEUE,        assemble order content8.Queue;
760$        GROUP,        keeptrack product,Permanent:counter(keeptrack product),Last:NEXT(761$);
761$        ASSIGN:       assemble order content8.NumberOut=assemble order content8.NumberOut +
1:NEXT(123$);

;
;    Model statements for module:  Record 49
;
123$        TALLY:        leadtime8 order,INT(order time8),1:NEXT(110$);
;
;    Model statements for module:  Process 33
;
147$        ASSIGN:       delay9.NumberIn=delay9.NumberIn + 1:
                     delay9.WIP=delay9.WIP+1;
763$        DELAY:        0.000000027777778,,VA;
810$        ASSIGN:       delay9.NumberOut=delay9.NumberOut + 1:
                     delay9.WIP=delay9.WIP-1:NEXT(126$);
;
;    Model statements for module:  Batch 44
;
126$        QUEUE,        assemble order content9.Queue;
813$        GROUP,        keeptrack product,Permanent:counter(keeptrack product),Last:NEXT(814$);
814$        ASSIGN:       assemble order content9.NumberOut=assemble order content9.NumberOut +
1:NEXT(125$);
;
;    Model statements for module:  Record 50
;
125$        TALLY:        leadtime9 order,INT(order time9),1:NEXT(110$);

;
;    Model statements for module:  Process 34
;
148$        ASSIGN:       delay10.NumberIn=delay10.NumberIn + 1:
                     delay10.WIP=delay10.WIP+1;
816$        DELAY:        0.000000027777778,,VA;
863$        ASSIGN:       delay10.NumberOut=delay10.NumberOut + 1:
                     delay10.WIP=delay10.WIP-1:NEXT(128$);
;
;    Model statements for module:  Batch 45
;
128$        QUEUE,        assemble order content10.Queue;
```

```
866$        GROUP,         keeptrack product,Permanent:counter(keeptrack product),Last:NEXT(867$);
867$        ASSIGN:        assemble order content10.NumberOut=assemble order content10.NumberOut +
1:NEXT(127$);
;
;    Model statements for module:  Record 51
;
127$        TALLY:         leadtime10 order,INT(order time10),1:NEXT(110$);
;
;    Model statements for module:  Process 35
;
149$        ASSIGN:        delay11.NumberIn=delay11.NumberIn + 1:
                    delay11.WIP=delay11.WIP+1;
869$        DELAY:         0.000000027777778,,VA;
916$        ASSIGN:        delay11.NumberOut=delay11.NumberOut + 1:
                    delay11.WIP=delay11.WIP-1:NEXT(130$);
;
;    Model statements for module:  Batch 46
;
130$        QUEUE,         assemble order content11.Queue;
919$        GROUP,         keeptrack product,Permanent:counter(keeptrack product),Last:NEXT(920$);
920$        ASSIGN:        assemble order content11.NumberOut=assemble order content11.NumberOut +
1:NEXT(129$);
;
;    Model statements for module:  Record 52
;
129$        TALLY:         leadtime11 order,INT(order time11),1:NEXT(110$);
;
;    Model statements for module:  Process 36
;
150$        ASSIGN:        delay12.NumberIn=delay12.NumberIn + 1:
                    delay12.WIP=delay12.WIP+1;
922$        DELAY:         0.000000027777778,,VA;
969$        ASSIGN:        delay12.NumberOut=delay12.NumberOut + 1:
                    delay12.WIP=delay12.WIP-1:NEXT(132$);
;
;    Model statements for module:  Batch 47
;
132$        QUEUE,         assemble order content12.Queue;
972$        GROUP,         keeptrack product,Permanent:counter(keeptrack product),Last:NEXT(973$);
973$        ASSIGN:        assemble order content12.NumberOut=assemble order content12.NumberOut +
1:NEXT(131$);
;
;    Model statements for module:  Record 53
;
131$        TALLY:         leadtime12 order,INT(order time12),1:NEXT(110$);
;
;    Model statements for module:  Process 37
;
151$        ASSIGN:        delay13.NumberIn=delay13.NumberIn + 1:
                    delay13.WIP=delay13.WIP+1;
```

```
975$        DELAY:       0.000000027777778,,VA;
1022$       ASSIGN:      delay13.NumberOut=delay13.NumberOut + 1:
                    delay13.WIP=delay13.WIP-1:NEXT(134$);
;
;    Model statements for module:  Batch 48
;
134$        QUEUE,       assemble order content13.Queue;
1025$       GROUP,       keeptrack product,Permanent:counter(keeptrack product),Last:NEXT(1026$);
1026$       ASSIGN:      assemble order content13.NumberOut=assemble order content13.NumberOut +
1:NEXT(133$);
;
;    Model statements for module:  Record 54
;
133$        TALLY:       leadtime13 order,INT(order time13),1:NEXT(110$);
;
;    Model statements for module:  Process 38
;
152$        ASSIGN:      delay14.NumberIn=delay14.NumberIn + 1:
                    delay14.WIP=delay14.WIP+1;
1028$       DELAY:       0.000000027777778,,VA;
1075$       ASSIGN:      delay14.NumberOut=delay14.NumberOut + 1:
                    delay14.WIP=delay14.WIP-1:NEXT(136$);
;
;    Model statements for module:  Batch 49
;
136$        QUEUE,       assemble order content14.Queue;
1078$       GROUP,       keeptrack product,Permanent:counter(keeptrack product),Last:NEXT(1079$);
1079$       ASSIGN:      assemble order content14.NumberOut=assemble order content14.NumberOut +
1:NEXT(135$);
;
;    Model statements for module:  Record 55
;
135$        TALLY:       leadtime14 order,INT(order time14),1:NEXT(110$);
;
;    Model statements for module:  Batch 1
;
11$         QUEUE,       assemble part1.Queue;
1080$       GROUP,       demand1,Permanent:17,Last:NEXT(1081$);
1081$       ASSIGN:      assemble part1.NumberOut=assemble part1.NumberOut + 1:NEXT(102$);
;
;    Model statements for module:  Process 5
;
102$        ASSIGN:      assembly line1.NumberIn=assembly line1.NumberIn + 1:
                    assembly line1.WIP=assembly line1.WIP+1;
1083$       DELAY:       SecondsToBaseTime(NORM(114,2)),,VA;
1130$       ASSIGN:      assembly line1.NumberOut=assembly line1.NumberOut + 1:
                    assembly line1.WIP=assembly line1.WIP-1:NEXT(16$);
;
;    Model statements for module:  Record 1
;
```

```
16$        COUNT:      countparttype1done,1:NEXT(46$);
;
;    Model statements for module:  Record 6
;
46$        TALLY:      leadtime1,INT(starttime1),1:NEXT(107$);
;
;    Model statements for module:  Batch 2
;
12$        QUEUE,      assemble part2.Queue;
1133$        GROUP,      demand2,Permanent:23,Last:NEXT(1134$);
1134$        ASSIGN:     assemble part2.NumberOut=assemble part2.NumberOut + 1:NEXT(103$);
;
;    Model statements for module:  Process 6
;
103$        ASSIGN:     assembly line2.NumberIn=assembly line2.NumberIn + 1:
                 assembly line2.WIP=assembly line2.WIP+1;
1136$        DELAY:      SecondsToBaseTime(NORM(114,2)),,VA;
1183$        ASSIGN:     assembly line2.NumberOut=assembly line2.NumberOut + 1:
                 assembly line2.WIP=assembly line2.WIP-1:NEXT(17$);
;
;    Model statements for module:  Record 2
;
17$        COUNT:      countparttype2done,1:NEXT(98$);
;
;    Model statements for module:  Record 7
;
98$        TALLY:      leadtime2,INT(starttime2),1:NEXT(107$);
;
;    Model statements for module:  Batch 3
;
13$        QUEUE,      assemble part3.Queue;
1186$        GROUP,      demand3,Permanent:25,Last:NEXT(1187$);
1187$        ASSIGN:     assemble part3.NumberOut=assemble part3.NumberOut + 1:NEXT(104$);
;
;    Model statements for module:  Process 7
;
104$        ASSIGN:     assembly line3.NumberIn=assembly line3.NumberIn + 1:
                 assembly line3.WIP=assembly line3.WIP+1;
1189$        DELAY:      SecondsToBaseTime(NORM(114,2)),,VA;
1236$        ASSIGN:     assembly line3.NumberOut=assembly line3.NumberOut + 1:
                 assembly line3.WIP=assembly line3.WIP-1:NEXT(18$);
;
;    Model statements for module:  Record 3
;
18$        COUNT:      countparttype3done,1:NEXT(99$);
;
;    Model statements for module:  Record 8
;
99$        TALLY:      leadtime3,INT(starttime3),1:NEXT(107$);
;
```

```
;    Model statements for module:  Batch 4
;
14$         QUEUE,        assemble part4.Queue;
1239$        GROUP,        demand4,Permanent:35,Last:NEXT(1240$);
1240$        ASSIGN:       assemble part4.NumberOut=assemble part4.NumberOut + 1:NEXT(105$);
;    Model statements for module:  Process 8
;
105$        ASSIGN:       assembly line4.NumberIn=assembly line4.NumberIn + 1:
                    assembly line4.WIP=assembly line4.WIP+1;
1242$        DELAY:        SecondsToBaseTime(NORM(114,2)),,VA;
1289$        ASSIGN:       assembly line4.NumberOut=assembly line4.NumberOut + 1:
                    assembly line4.WIP=assembly line4.WIP-1:NEXT(19$);
;
;    Model statements for module:  Record 4
;
19$         COUNT:        countparttype4done,1:NEXT(100$);
;
;    Model statements for module:  Record 9
;
100$        TALLY:        leadtime4,INT(starttime4),1:NEXT(107$);
;
;    Model statements for module:  Process 1
;
6$          ASSIGN:       purchase bit.NumberIn=purchase bit.NumberIn + 1:
                    purchase bit.WIP=purchase bit.WIP+1;
1295$        QUEUE,        purchase bit.Queue;
1294$        SEIZE,        2,VA:
                    purchase,1:NEXT(1293$);
1293$        DELAY:        SecondsToBaseTime(process time (part type, component)),,VA;
1292$        RELEASE:      purchase,1;
1340$        ASSIGN:       purchase bit.NumberOut=purchase bit.NumberOut + 1:
                    purchase bit.WIP=purchase bit.WIP-1:NEXT(10$);
;
;    Model statements for module:  Process 2
;
7$          ASSIGN:       injection process.NumberIn=injection process.NumberIn + 1:
                    injection process.WIP=injection process.WIP+1;
1346$        QUEUE,        injection process.Queue;
1345$        SEIZE,        2,VA:
                    injection,1:NEXT(1344$);
1344$        DELAY:        SecondsToBaseTime(process time (part type, component)),,VA;
1343$        RELEASE:      injection,1;
1391$        ASSIGN:       injection process.NumberOut=injection process.NumberOut + 1:
                    injection process.WIP=injection process.WIP-1:NEXT(10$);
;
;    Model statements for module:  Process 4
;
9$          ASSIGN:       diecasting process.NumberIn=diecasting process.NumberIn + 1:
                    diecasting process.WIP=diecasting process.WIP+1;
1397$        QUEUE,        diecasting process.Queue;
```

```
1396$       SEIZE,      2,VA:
                  diecasting,1:NEXT(1395$);
1395$       DELAY:      SecondsToBaseTime(process time (part type, component)),,VA;
1394$       RELEASE:     diecasting,1;
1442$       ASSIGN:      diecasting process.NumberOut=diecasting process.NumberOut + 1:
                  diecasting process.WIP=diecasting process.WIP-1:NEXT(10$);
;
;    Model statements for module:  Assign 30
;
25$        ASSIGN:      cdemand1=cdemand1+1:
                  demand1=cdemand1:
                  starttime1=TNOW:NEXT(21$);
;
;    Model statements for module:  Separate 6
;
21$        DUPLICATE,    100 - 0:
                  16,1447$,0:NEXT(1446$);

1446$       ASSIGN:      create components.NumberOut Orig=create components.NumberOut Orig +
1:NEXT(22$);
1447$       ASSIGN:      create components.NumberOut Dup=create components.NumberOut Dup +
1:NEXT(22$);
;
;    Model statements for module:  Assign 28
;
22$        ASSIGN:      say=say + 1:
                  component=say:NEXT(23$);
;
;    Model statements for module:  Decide 14
;
23$        BRANCH,      1:
                  If,component==17,1448$,Yes:
                  Else,1449$,Yes;
1448$       ASSIGN:      counter 0?.NumberOut True=counter 0?.NumberOut True + 1:NEXT(24$);
1449$       ASSIGN:      counter 0?.NumberOut False=counter 0?.NumberOut False + 1:NEXT(1$);
;
;    Model statements for module:  Assign 29
;
24$        ASSIGN:      say=0:NEXT(1$);
;
;    Model statements for module:  Decide 3
;
1$         BRANCH,      1:
                  If,component == 2 .or. component == 3 .or. component == 6,7$,Yes:
                  If,
                  component == 1 .OR. component == 4 .OR. component == 5 .OR. component == 8 .OR.
component == 10 .OR. component == 14 .OR. component == 15 .OR. component == 16.OR. component ==
17,
                  6$,Yes:
                  Else,8$,Yes;
```

```
;
;    Model statements for module:  Assign 33
;
30$         ASSIGN:       cdemand2=cdemand2+1:
                          demand2=cdemand2:
                          starttime2=TNOW:NEXT(26$);
;
;    Model statements for module:  Separate 7
;
26$         DUPLICATE,     100 - 0:
                          22,1454$,0:NEXT(1453$);
1453$        ASSIGN:       create components part2.NumberOut Orig=create components part2.NumberOut
Orig + 1:NEXT(27$);
1454$        ASSIGN:       create components part2.NumberOut Dup=create components part2.NumberOut
Dup + 1:NEXT(27$);
;
;    Model statements for module:  Assign 31
;
27$         ASSIGN:       say=say + 1:
                          component=say:NEXT(28$);
;
;    Model statements for module:  Decide 15
;
28$         BRANCH,       1:
                          If,component==23,1455$,Yes:
                          Else,1456$,Yes;
1455$        ASSIGN:       counter 0?2.NumberOut True=counter 0?2.NumberOut True + 1:NEXT(29$);
1456$        ASSIGN:       counter 0?2.NumberOut False=counter 0?2.NumberOut False + 1:NEXT(2$);
;
;    Model statements for module:  Assign 32
;
29$         ASSIGN:       say=0:NEXT(2$);
;
;
;    Model statements for module:  Decide 6
;
2$          BRANCH,       1:
                          If,
                          component == 1 .or. component == 4 .or. component == 5 .or. component == 8 .or.
component == 10 .or. component == 11 .or. component == 15 .or. component == 16 .or. component == 17
.or. component == 18 .or. component == 19 .or. component == 20 .or. component == 21 .or. component ==
22 .or. component == 23,
                          6$,Yes:
                          If,component == 2 .or. component == 3 .or. component == 6,7$,Yes:
                          Else,8$,Yes;
;
;    Model statements for module:  Assign 36
;
35$         ASSIGN:       cdemand3=cdemand3+1:
                          demand3=cdemand3:
```

```
                    starttime3=TNOW:NEXT(31$);
;
;   Model statements for module:  Separate 8
;
31$         DUPLICATE,     100 - 0:
                    24,1461$,0:NEXT(1460$);
1460$       ASSIGN:        create components3.NumberOut Orig=create components3.NumberOut Orig +
1:NEXT(32$);
1461$       ASSIGN:        create components3.NumberOut Dup=create components3.NumberOut Dup +
1:NEXT(32$);
;
;   Model statements for module:  Assign 34
;
32$         ASSIGN:        say=say + 1:
                    component=say:NEXT(33$);
;
;
;   Model statements for module:  Decide 16
;
33$         BRANCH,        1:
                    If,component==25,1462$,Yes:
                    Else,1463$,Yes;
1462$       ASSIGN:        counter 0?3.NumberOut True=counter 0?3.NumberOut True + 1:NEXT(34$);
1463$       ASSIGN:        counter 0?3.NumberOut False=counter 0?3.NumberOut False + 1:NEXT(3$);
;
;   Model statements for module:  Assign 35
;
34$         ASSIGN:        say=0:NEXT(3$);
;
;   Model statements for module:  Decide 8
;
3$          BRANCH,        1:
                    If,
                    component == 1 .or. component == 2 .or. component == 5 .or. component == 7 .or.
component == 10 .or. component == 12 .or. component == 13 .or. component == 17 .or. component == 18
.or. component == 19 .or. component == 20 .or. component == 21 .or. component == 22 .or. component ==
23 .or. component == 24 .or. component == 25,
                    6$,Yes:
                    If,component == 3 .or. component == 4 .or. component == 8,7$,Yes:
                    Else,8$,Yes;
;
;   Model statements for module:  Assign 39
;
40$         ASSIGN:        cdemand4=cdemand4+1:
                    demand4=cdemand4:
                    starttime4=TNOW:NEXT(36$);
;
;   Model statements for module:  Separate 9
;
36$         DUPLICATE,     100 - 0:
```

```
                    34,1468$,0:NEXT(1467$);
1467$     ASSIGN:     create components4.NumberOut Orig=create components4.NumberOut Orig +
1:NEXT(37$);
1468$     ASSIGN:     create components4.NumberOut Dup=create components4.NumberOut Dup +
1:NEXT(37$);
;
;   Model statements for module:  Assign 37
;
37$       ASSIGN:     say=say + 1:
                    component=say:NEXT(38$);
;
;   Model statements for module:  Decide 17
;
38$       BRANCH,     1:
                    If,component==35,1469$,Yes:
                    Else,1470$,Yes;
1469$     ASSIGN:     counter 0?4.NumberOut True=counter 0?4.NumberOut True + 1:NEXT(39$);
1470$     ASSIGN:     counter 0?4.NumberOut False=counter 0?4.NumberOut False + 1:NEXT(4$);
;
;   Model statements for module:  Assign 38
;
39$       ASSIGN:     say=0:NEXT(4$);
;
;   Model statements for module:  Decide 10
;
4$        BRANCH,     1:
                    If,
                    component == 1 .or. component == 6 .or. component == 7 .or. component == 8 .or.
component == 9 .or. component == 13 .or. component == 15 .or. component == 18 .or. component == 20 .or.
component == 21 .or. component == 25 .or. component == 26 .or. component == 27 .or. component == 28
.or. component == 29 .or. component == 33 .or. component == 34 .or. component == 35,
                    6$,Yes:
                    If,
                    component == 2 .or. component == 3 .or. component == 10 .or. component == 11 .or.
component == 12 .or. component == 16 .or. component == 30 .or. component == 31 .or. component == 32,
                    7$,Yes:
                    If,component == 14,9$,Yes:
                    Else,8$,Yes;


;
;   Model statements for module:  Assign 82
;
97$       ASSIGN:     order time15=TNOW:
                    part type=5:
                    counter(keeptrack product)=counter(keeptrack product) + 1:NEXT(0$);
;
;
;   Model statements for module:  Separate 11
;
48$       DUPLICATE,  100 - 50:
```

```
                    NORM(332,32),1475$,50:NEXT(1474$);
1474$      ASSIGN:      p1 drivers.NumberOut Orig=p1 drivers.NumberOut Orig + 1:NEXT(49$);
1475$      ASSIGN:      p1 drivers.NumberOut Dup=p1 drivers.NumberOut Dup + 1:NEXT(49$);
;
;   Model statements for module:  Assign 58
;
49$        ASSIGN:      order time1=TNOW:
 ÿÿ                 ÿÿÿÿart type=1:
                ÿÿÿÿ      counter(keeptrack product)=counter(keeptrack product) + 1:NEXT(0$);
;
;   Model statements for module:  Separate 12
;
50$        DUPLICATE,    100 - 50:
                    NORM(332,32),1478$,50:NEXT(1477$);
1477$      ASSIGN:      p2 drivers.NumberOut Orig=p2 drivers.NumberOut Orig + 1:NEXT(51$);
1478$      ASSIGN:      p2 drivers.NumberOut Dup=p2 drivers.NumberOut Dup + 1:NEXT(51$);
;
;   Model statements for module:  Assign 59
;
51$        ASSIGN:      order time2=TNOW:
                    part type=2:
                    counter(keeptrack product)=counter(keeptrack product) + 1:NEXT(0$);
;
;   Model statements for module:  Separate 13
;
52$        DUPLICATE,    100 - 50:
                    NORM(332,32),1481$,50:NEXT(1480$);
1480$      ASSIGN:      p3 drivers.NumberOut Orig=p3 drivers.NumberOut Orig + 1:NEXT(53$);
1481$      ASSIGN:      p3 drivers.NumberOut Dup=p3 drivers.NumberOut Dup + 1:NEXT(53$);
;
;   Model statements for module:  Assign 60
;
53$        ASSIGN:      order time3=TNOW:
                    part type=3:
                    counter(keeptrack product)=counter(keeptrack product) + 1:NEXT(0$);
;
;   Model statements for module:  Separate 14
;
54$        DUPLICATE,    100 - 50:
                    NORM(332,32),1484$,50:NEXT(1483$);
1483$      ASSIGN:      p4 drivers.NumberOut Orig=p4 drivers.NumberOut Orig + 1:NEXT(55$);
1484$      ASSIGN:      p4 drivers.NumberOut Dup=p4 drivers.NumberOut Dup + 1:NEXT(55$);
;
;   Model statements for module:  Assign 61
;
55$        ASSIGN:      order time4=TNOW:
                    part type=4:
                    counter(keeptrack product)=counter(keeptrack product) + 1:NEXT(0$);
;
;   Model statements for module:  Separate 15
```

```
;
56$         DUPLICATE,    100 - 50:
                    NORM(332,32),1487$,50:NEXT(1486$);
1486$        ASSIGN:       p5 drivers.NumberOut Orig=p5 drivers.NumberOut Orig + 1:NEXT(57$);
1487$        ASSIGN:       p5 drivers.NumberOut Dup=p5 drivers.NumberOut Dup + 1:NEXT(57$);
;
;     Model statements for module:  Assign 62
;
57$         ASSIGN:       order time5=TNOW:
                    part type=5:
                    counter(keeptrack product)=counter(keeptrack product) + 1:NEXT(0$);
;
;     Model statements for module:  Separate 16
;
58$         DUPLICATE,    100 - 50:
                    NORM(166,16),1490$,50:NEXT(1489$);
1489$        ASSIGN:       p2 out of order content p1 and p2.NumberOut Orig=
                    p2 out of order content p1 and p2.NumberOut Orig + 1:NEXT(59$);
1490$        ASSIGN:       p2 out of order content p1 and p2.NumberOut Dup=p2 out of order content p1 and
p2.NumberOut Dup + 1
                    :NEXT(61$);
;
;     Model statements for module:  Separate 17
;
59$         DUPLICATE,    100 - 50:
                    NORM(166,16),1493$,50:NEXT(1492$);
1492$        ASSIGN:       p1 out of order content p1 and p2.NumberOut Orig=
                    p1 out of order content p1 and p2.NumberOut Orig + 1:NEXT(60$);
1493$        ASSIGN:       p1 out of order content p1 and p2.NumberOut Dup=p1 out of order content p1 and
p2.NumberOut Dup + 1
                    :NEXT(60$);
;
;     Model statements for module:  Assign 63
;
60$         ASSIGN:       order time6=TNOW:
                    part type=1:
                    counter(keeptrack product)=counter(keeptrack product) + 1:NEXT(0$);
;
;     Model statements for module:  Assign 64
;
61$         ASSIGN:       order time6=TNOW:
                    part type=2:
                    counter(keeptrack product)=counter(keeptrack product) + 1:NEXT(0$);
;
;     Model statements for module:  Separate 18
;
62$         DUPLICATE,    100 - 50:
                    NORM(166,16),1496$,50:NEXT(1495$);
1495$        ASSIGN:       p3 out of order content p1 and p3.NumberOut Orig=
                    p3 out of order content p1 and p3.NumberOut Orig + 1:NEXT(63$);
```

```
1496$        ASSIGN:       p3 out of order content p1 and p3.NumberOut Dup=p3 out of order content p1 and
p3.NumberOut Dup + 1
                     :NEXT(65$);
;
;    Model statements for module:  Separate 19
;
63$          DUPLICATE,     100 - 50:
                     NORM(166,16),1499$,50:NEXT(1498$);
1498$        ASSIGN:       p1 out of order content p1 and p3.NumberOut Orig=
                     p1 out of order content p1 and p3.NumberOut Orig + 1:NEXT(64$);
1499$        ASSIGN:       p1 out of order content p1 and p3.NumberOut Dup=p1 out of order content p1 and
p3.NumberOut Dup + 1
                     :NEXT(64$);
;
;    Model statements for module:  Assign 65
;
64$          ASSIGN:       order time7=TNOW:
                     part type=1:
                     counter(keeptrack product)=counter(keeptrack product) + 1:NEXT(0$);
;
;    Model statements for module:  Assign 66
;
65$          ASSIGN:       order time7=TNOW:
                     part type=3:
                     counter(keeptrack product)=counter(keeptrack product) + 1:NEXT(0$);
;
;    Model statements for module:  Separate 20
;
66$          DUPLICATE,     100 - 50:
                     NORM(166,16),1502$,50:NEXT(1501$);
1501$        ASSIGN:       p4 out of order content p1 and p4.NumberOut Orig=
                     p4 out of order content p1 and p4.NumberOut Orig + 1:NEXT(67$);
1502$        ASSIGN:       p4 out of order content p1 and p4.NumberOut Dup=p4 out of order content p1 and
p4.NumberOut Dup + 1
                     :NEXT(69$);
;
;    Model statements for module:  Separate 21
;
67$          DUPLICATE,     100 - 50:
                     NORM(166,16),1505$,50:NEXT(1504$);

1504$        ASSIGN:       p1 out of order content p1 and p4.NumberOut Orig=
                     p1 out of order content p1 and p4.NumberOut Orig + 1:NEXT(68$);
1505$        ASSIGN:       p1 out of order content p1 and p4.NumberOut Dup=p1 out of order content p1 and
p4.NumberOut Dup + 1
                     :NEXT(68$);
;
;    Model statements for module:  Assign 67
;
68$          ASSIGN:       order time8=TNOW:
```

```
                        part type=1:
                        counter(keeptrack product)=counter(keeptrack product) + 1:NEXT(0$);
;
;    Model statements for module:  Assign 68
;
69$         ASSIGN:      order time8=TNOW:
                        part type=4:
                        counter(keeptrack product)=counter(keeptrack product) + 1:NEXT(0$);
;
;    Model statements for module:  Separate 22
;
70$         DUPLICATE,    100 - 50:
                        NORM(166,16),1508$,50:NEXT(1507$);
1507$        ASSIGN:      p5 out of order content p1 and p5.NumberOut Orig=
                        p5 out of order content p1 and p5.NumberOut Orig + 1:NEXT(71$);
1508$        ASSIGN:      p5 out of order content p1 and p5.NumberOut Dup=p5 out of order content p1 and
p5.NumberOut Dup + 1
                        :NEXT(73$);
;
;    Model statements for module:  Separate 23
;
71$         DUPLICATE,    100 - 50:
                        NORM(166,16),1511$,50:NEXT(1510$);
1510$        ASSIGN:      p1 out of order content p1 and p5.NumberOut Orig=
                        p1 out of order content p1 and p5.NumberOut Orig + 1:NEXT(72$);
1511$        ASSIGN:      p1 out of order content p1 and p5.NumberOut Dup=p1 out of order content p1 and
p5.NumberOut Dup + 1
                        :NEXT(72$);
;
;    Model statements for module:  Assign 69
;
72$         ASSIGN:      order time9=TNOW:
                        part type=1:
                        counter(keeptrack product)=counter(keeptrack product) + 1:NEXT(0$);
;
;    Model statements for module:  Assign 70
;
73$         ASSIGN:      order time9=TNOW:
                        part type=5:
                        counter(keeptrack product)=counter(keeptrack product) + 1:NEXT(0$);
;
;    Model statements for module:  Separate 24
;
74$         DUPLICATE,    100 - 50:
                        NORM(166,16),1514$,50:NEXT(1513$);
1513$        ASSIGN:      p3 out of order content p2 and p3.NumberOut Orig=
                        p3 out of order content p2 and p3.NumberOut Orig + 1:NEXT(75$);
1514$        ASSIGN:      p3 out of order content p2 and p3.NumberOut Dup=p3 out of order content p2 and
p3.NumberOut Dup + 1
                        :NEXT(77$);
```

```
;
;     Model statements for module:  Separate 25
;
75$          DUPLICATE,     100 - 50:
                        NORM(166,16),1517$,50:NEXT(1516$);
1516$          ASSIGN:       p2 out of order content p2 and p3.NumberOut Orig=
                        p2 out of order content p2 and p3.NumberOut Orig + 1:NEXT(76$);
1517$          ASSIGN:       p2 out of order content p2 and p3.NumberOut Dup=p2 out of order content p2 and
p3.NumberOut Dup + 1
                        :NEXT(76$);
;
;     Model statements for module:  Assign 71
;
76$          ASSIGN:       order time10=TNOW:
                        part type=2:
                        counter(keeptrack product)=counter(keeptrack product) + 1:NEXT(0$);
;
;     Model statements for module:  Assign 72
;
77$          ASSIGN:       order time10=TNOW:
                        part type=3:
                        counter(keeptrack product)=counter(keeptrack product) + 1:NEXT(0$);
;
;     Model statements for module:  Separate 26
;
78$          DUPLICATE,     100 - 50:
                        NORM(166,16),1520$,50:NEXT(1519$);
1519$          ASSIGN:       p4 out of order content p2 and p4.NumberOut Orig=
                        p4 out of order content p2 and p4.NumberOut Orig + 1:NEXT(79$);
1520$          ASSIGN:       p4 out of order content p2 and p4.NumberOut Dup=p4 out of order content p2 and
p4.NumberOut Dup + 1
                        :NEXT(81$);
;
;     Model statements for module:  Separate 27
;
79$          DUPLICATE,     100 - 50:
                        NORM(166,16),1523$,50:NEXT(1522$);
1522$          ASSIGN:       p2 out of order content p2 and p4.NumberOut Orig=
                        p2 out of order content p2 and p4.NumberOut Orig + 1:NEXT(80$);
1523$          ASSIGN:       p2 out of order content p2 and p4.NumberOut Dup=p2 out of order content p2 and
p4.NumberOut Dup + 1
                        :NEXT(80$);
;
;     Model statements for module:  Assign 73
;
80$          ASSIGN:       order time11=TNOW:
                        part type=2:
                        counter(keeptrack product)=counter(keeptrack product) + 1:NEXT(0$);
;
;     Model statements for module:  Assign 74
```

```
;
81$         ASSIGN:      order time11=TNOW:
                         part type=4:
                         counter(keeptrack product)=counter(keeptrack product) + 1:NEXT(0$);
;
;   Model statements for module:  Separate 28
;
82$         DUPLICATE,    100 - 50:
                         NORM(166,16),1526$,50:NEXT(1525$);
1525$        ASSIGN:      p5 out of order content p2 and p5.NumberOut Orig=
                         p5 out of order content p2 and p5.NumberOut Orig + 1:NEXT(83$);
1526$        ASSIGN:      p5 out of order content p2 and p5.NumberOut Dup=p5 out of order content p2 and
p5.NumberOut Dup + 1
                         :NEXT(85$);
;
;   Model statements for module:  Separate 29
;
83$         DUPLICATE,    100 - 50:
                         NORM(166,16),1529$,50:NEXT(1528$);
1528$        ASSIGN:      p2 out of order content p2 and p5.NumberOut Orig=
                         p2 out of order content p2 and p5.NumberOut Orig + 1:NEXT(84$);
1529$        ASSIGN:      p2 out of order content p2 and p5.NumberOut Dup=p2 out of order content p2 and
p5.NumberOut Dup + 1
                         :NEXT(84$);
;
;   Model statements for module:  Assign 75
;
84$         ASSIGN:      order time12=TNOW:
                         part type=2:
                         counter(keeptrack product)=counter(keeptrack product) + 1:NEXT(0$);
;
;   Model statements for module:  Assign 76
;
85$         ASSIGN:      order time12=TNOW:
                         part type=5:
                         counter(keeptrack product)=counter(keeptrack product) + 1:NEXT(0$);
;
;   Model statements for module:  Separate 30
;
86$         DUPLICATE,    100 - 50:
                         NORM(166,16),1532$,50:NEXT(1531$);
1531$        ASSIGN:      p4 out of order content p3 and p4.NumberOut Orig=
                         p4 out of order content p3 and p4.NumberOut Orig + 1:NEXT(87$);
1532$        ASSIGN:      p4 out of order content p3 and p4.NumberOut Dup=p4 out of order content p3 and
p4.NumberOut Dup + 1
                         :NEXT(89$);
;
;   Model statements for module:  Separate 31
;
87$         DUPLICATE,    100 - 50:
```

```
                    NORM(166,16),1535$,50:NEXT(1534$);
1534$       ASSIGN:     p3 out of order content p3 and p4.NumberOut Orig=
                    p3 out of order content p3 and p4.NumberOut Orig + 1:NEXT(88$);
1535$       ASSIGN:     p3 out of order content p3 and p4.NumberOut Dup=p3 out of order content p3 and
p4.NumberOut Dup + 1
                    :NEXT(88$);
;
;    Model statements for module:  Assign 77
;
88$         ASSIGN:     order time13=TNOW:
                    part type=3:
                    counter(keeptrack product)=counter(keeptrack product) + 1:NEXT(0$);
;
;    Model statements for module:  Assign 78
;
89$         ASSIGN:     order time13=TNOW:
                    part type=4:
                    counter(keeptrack product)=counter(keeptrack product) + 1:NEXT(0$);
;
;    Model statements for module:  Separate 32
;
90$         DUPLICATE,    100 - 50:
                    NORM(166,16),1538$,50:NEXT(1537$);
1537$       ASSIGN:     p5 out of order content p3 and p5.NumberOut Orig=
                    p5 out of order content p3 and p5.NumberOut Orig + 1:NEXT(91$);
1538$       ASSIGN:     p5 out of order content p3 and p5.NumberOut Dup=p5 out of order content p3 and
p5.NumberOut Dup + 1
                    :NEXT(93$);
;
;    Model statements for module:  Separate 33
;
91$         DUPLICATE,    100 - 50:
                    NORM(166,16),1541$,50:NEXT(1540$);
1540$       ASSIGN:     p3 out of order content p3 and p5.NumberOut Orig=
                    p3 out of order content p3 and p5.NumberOut Orig + 1:NEXT(92$);
1541$       ASSIGN:     p3 out of order content p3 and p5.NumberOut Dup=p3 out of order content p3 and
p5.NumberOut Dup + 1
                    :NEXT(92$);


;
;    Model statements for module:  Assign 79
;
92$         ASSIGN:     order time14=TNOW:
                    part type=3:
                    counter(keeptrack product)=counter(keeptrack product) + 1:NEXT(0$);
;
;    Model statements for module:  Assign 80
;
93$         ASSIGN:     order time14=TNOW:
                    part type=5:
```

counter(keeptrack product)=counter(keeptrack product) + 1:NEXT(0$);

Appendix C

T-tests

make-to-order vs. make-to-stock ORDER LEAD TIMES

Paired T-test and CI: C14, C15

Paired T for C14 - C15

	N	Mean	StDev	SE Mean
C14	15	11.13	4.72	1.22
C15	15	2.88	0.51	0.13
Difference	15	8.26	4.83	1.25

95% CI for mean difference: (5.58, 10.93)
T-test of mean difference = 0 (vs not = 0): T-Value = 6.63 P-Value = 0.000

make-to-order vs. make-to-stock ORDER COSTS

Paired T-test and CI: C1, C2

Paired T for C1 - C2

	N	Mean	StDev	SE Mean
C1	15	54.99	22.64	5.85
C2	15	90.61	1.98	0.51
Difference	15	-35.62	21.57	5.57

95% CI for mean difference: (-47.56, -23.68)
T-test of mean difference = 0 (vs not = 0): T-Value = -6.40 P-Value = 0.000

120K Demand make-to-order vs. make-to-stock ORDER LEAD TIMES

Paired T-test and CI: C1, C2

Paired T for C1 - C2

	N	Mean	StDev	SE Mean
C1	15	18.09	4.69	1.21
C2	15	3.60	0.64	0.17
Difference	15	14.49	4.70	1.21

95% CI for mean difference: (11.89, 17.10)
T-test of mean difference = 0 (vs not = 0): T-Value = 11.93 P-Value = 0.000

120K Demand make-to-order vs. make-to-stock ORDER COSTS

Paired T-test and CI: C3, C4

Paired T for C3 - C4

	N	Mean	StDev	SE Mean
C3	15	77.29	22.82	5.89
C4	15	92.61	2.72	0.70
Difference	15	-15.32	21.44	5.54

95% CI for mean difference: (-27.20, -3.45)
T-test of mean difference = 0 (vs not = 0): T-Value = -2.77 P-Value = 0.015

100 K Demand make-to-order vs. 120K demand make-to-order ORDER LEAD TIMES

Paired T-test and CI: C5, C6

Paired T for C5 - C6

	N	Mean	StDev	SE Mean
C5	15	11.13	4.72	1.22
C6	15	18.09	4.69	1.21
Difference	15	-6.96	4.14	1.07

95% CI for mean difference: (-9.25, -4.67)
T-test of mean difference = 0 (vs not = 0): T-Value = -6.52 P-Value = 0.000

100 K Demand make-to-stock vs. 120K demand make-to-stock ORDER LEAD TIMES

Paired T-test and CI: C7, C8

Paired T for C7 - C8

	N	Mean	StDev	SE Mean
C7	15	2.876	0.505	0.130
C8	15	3.598	0.644	0.166
Difference	15	-0.722	0.515	0.133

95% CI for mean difference: (-1.007, -0.437)
T-test of mean difference = 0 (vs not = 0): T-Value = -5.43 P-Value = 0.000

100 K Demand make-to-order vs. 120K demand make-to-order ORDER COSTS

Paired T-test and CI: C9, C10

Paired T for C9 - C10

	N	Mean	StDev	SE Mean
C9	15	54.99	22.64	5.85
C10	15	77.29	22.82	5.89

	N	Mean	StDev	SE Mean
Difference	15	-22.29	12.61	3.26

95% CI for mean difference: (-29.28, -15.31)
T-test of mean difference = 0 (vs not = 0): T-Value = -6.85 P-Value = 0.000

100 K Demand make-to-stock vs. 120K demand make-to-stock ORDER COSTS

Paired T-test and CI: C11, C12

Paired T for C11 - C12

	N	Mean	StDev	SE Mean
C11	15	90.612	1.985	0.512
C12	15	92.609	2.722	0.703
Difference	15	-1.997	1.505	0.389

95% CI for mean difference: (-2.831, -1.164)
T-test of mean difference = 0 (vs not = 0): T-Value = -5.14 P-Value = 0.000

80K Demand make-to-order vs. make-to-stock ORDER LEAD TIMES

Paired T-test and CI: C1, C2

Paired T for C1 - C2

	N	Mean	StDev	SE Mean
C1	15	6.748	2.316	0.598
C2	15	2.382	0.574	0.148
Difference	15	4.366	2.644	0.683

95% CI for mean difference: (2.902, 5.831)
T-test of mean difference = 0 (vs not = 0): T-Value = 6.40 P-Value = 0.000

80K Demand make-to-order vs. make-to-stock ORDER COSTS

Paired T-test and CI: C3, C4

Paired T for C3 - C4

	N	Mean	StDev	SE Mean
C3	15	39.73	13.07	3.37
C4	15	89.25	1.86	0.48
Difference	15	-49.52	12.54	3.24

95% CI for mean difference: (-56.47, -42.58)
T-test of mean difference = 0 (vs not = 0): T-Value = -15.30 P-Value = 0.000

100 K make-to-order vs. 80K make-to-order ORDER LEAD TIMES

Paired T-test and CI: C5, C6

Paired T for C5 - C6

	N	Mean	StDev	SE Mean
C5	15	11.13	4.72	1.22
C6	15	6.75	2.32	0.60
Difference	15	4.383	3.029	0.782

95% CI for mean difference: (2.706, 6.060)
T-test of mean difference = 0 (vs not = 0): T-Value = 5.60 P-Value = 0.000

100 K make-to-stock vs. 80K make-to-stock ORDER LEAD TIMES

Paired T-test and CI: C7, C8

Paired T for C7 - C8

	N	Mean	StDev	SE Mean
C7	15	2.876	0.505	0.130
C8	15	2.382	0.574	0.148
Difference	15	0.494	0.417	0.108

95% CI for mean difference: (0.263, 0.725)
T-test of mean difference = 0 (vs not = 0): T-Value = 4.59 P-Value = 0.000

100 K make-to-order vs. 80K make-to-order ORDER COSTS

Paired T-test and CI: C9, C10

Paired T for C9 - C10

	N	Mean	StDev	SE Mean
C9	15	54.99	22.64	5.85
C10	15	39.73	13.07	3.37
Difference	15	15.27	11.63	3.00

95% CI for mean difference: (8.82, 21.71)
T-test of mean difference = 0 (vs not = 0): T-Value = 5.08 P-Value = 0.000

100 K make-to-stock vs. 80K make-to-stock ORDER COSTS

Paired T-test and CI: C11, C12

Paired T for C11 - C12

	N	Mean	StDev	SE Mean
C11	15	90.612	1.985	0.512

	N	Mean	StDev	SE Mean
C12	15	89.250	1.858	0.480
Difference	15	1.362	1.156	0.298

95% CI for mean difference: (0.722, 2.002)
T-test of mean difference = 0 (vs not = 0): T-Value = 4.56 P-Value = 0.000

Extra Injection make-to-order vs. make-to-stock ORDER LEAD TIMES

Paired T-test and CI: C1, C2

Paired T for C1 - C2

	N	Mean	StDev	SE Mean
C1	15	8.443	3.488	0.900
C2	15	2.876	0.505	0.130
Difference	15	5.567	3.686	0.952

95% CI for mean difference: (3.525, 7.608)
T-test of mean difference = 0 (vs not = 0): T-Value = 5.85 P-Value = 0.000

Extra Injection make-to-order vs. make-to-stock ORDER COSTS

Paired T-test and CI: C3, C4

Paired T for C3 - C4

	N	Mean	StDev	SE Mean
C3	15	46.11	18.76	4.84
C4	15	90.61	1.98	0.51
Difference	15	-44.50	17.50	4.52

95% CI for mean difference: (-54.19, -34.81)
T-test of mean difference = 0 (vs not = 0): T-Value = -9.85 P-Value = 0.000

100 K make-to-order vs. Extra Injection make-to-order ORDER LEAD TIMES

Paired T-test and CI: C5, C6

Paired T for C5 - C6

	N	Mean	StDev	SE Mean
C5	15	11.13	4.72	1.22
C6	15	8.44	3.49	0.90
Difference	15	2.688	2.676	0.691

95% CI for mean difference: (1.207, 4.170)
T-test of mean difference = 0 (vs not = 0): T-Value = 3.89 P-Value = 0.002

100 K make-to-stock vs. Extra Injection make-to-stock ORDER LEAD TIMES

Paired T-test and CI: C7, C8

Paired T for C7 - C8

	N	Mean	StDev	SE Mean
C7	15	2.876	0.505	0.130
C8	15	2.876	0.505	0.130
Difference	15	0.000000	0.000000	0.000000

95% CI for mean difference: (0.000000, 0.000000)
T-test of mean difference = 0 (vs not = 0): T-Value = * P-Value = *

* NOTE * All values in column are identical.

100 K make-to-order vs. Extra Injection make-to-order ORDER COSTS

Paired T-test and CI: C9, C10

Paired T for C9 - C10

	N	Mean	StDev	SE Mean
C9	15	54.99	22.64	5.85
C10	15	46.11	18.76	4.84
Difference	15	8.88	9.21	2.38

95% CI for mean difference: (3.78, 13.99)
T-test of mean difference = 0 (vs not = 0): T-Value = 3.73 P-Value = 0.002

100 K make-to-stock vs. Extra Injection make-to-stock ORDER COSTS

Paired T-test and CI: C11, C12

Paired T for C11 - C12

	N	Mean	StDev	SE Mean
C11	15	90.612	1.985	0.512
C12	15	90.612	1.985	0.512
Difference	15	0.000000	0.000000	0.000000

95% CI for mean difference: (0.000000, 0.000000)
T-test of mean difference = 0 (vs not = 0): T-Value = * P-Value = *

* NOTE * All values in column are identical.

Extra Die-Casting make-to-order vs. make-to-stock ORDER LEAD TIMES

Paired T-test and CI: C1, C2

Paired T for C1 - C2

	N	Mean	StDev	SE Mean
C1	15	9.123	3.443	0.889
C2	15	2.876	0.505	0.130
Difference	15	6.247	3.692	0.953

95% CI for mean difference: (4.202, 8.292)
T-test of mean difference = 0 (vs not = 0): T-Value = 6.55 P-Value = 0.000

Extra Die-casting make-to-order vs. make-to-stock ORDER COSTS

Paired T-test and CI: C3, C4

Paired T for C3 - C4

	N	Mean	StDev	SE Mean
C3	15	48.22	18.56	4.79
C4	15	90.61	1.98	0.51
Difference	15	-42.39	17.54	4.53

95% CI for mean difference: (-52.10, -32.68)
T-test of mean difference = 0 (vs not = 0): T-Value = -9.36 P-Value = 0.000

100 K make-to-order vs. Extra Die-Casting make-to-order ORDER LEAD TIMES

Paired T-test and CI: C5, C6

Paired T for C5 - C6

	N	Mean	StDev	SE Mean
C5	15	11.13	4.72	1.22
C6	15	9.12	3.44	0.89
Difference	15	2.008	2.926	0.755

95% CI for mean difference: (0.388, 3.629)
T-test of mean difference = 0 (vs not = 0): T-Value = 2.66 P-Value = 0.019

100 K make-to-stock vs. Extra Die-Casting make-to-stock ORDER LEAD TIMES

Paired T-test and CI: C7, C8

Paired T for C7 - C8

	N	Mean	StDev	SE Mean
C7	15	2.876	0.505	0.130
C8	15	2.876	0.505	0.130

Difference	15	0.000000	0.000000	0.000000

95% CI for mean difference: (0.000000, 0.000000)
T-test of mean difference = 0 (vs not = 0): T-Value = * P-Value = *

* NOTE * All values in column are identical.

100 K make-to-order vs. Extra Injection make-to-order ORDER COSTS

Paired T-test and CI: C9, C10

Paired T for C9 - C10

	N	Mean	StDev	SE Mean
C9	15	54.99	22.64	5.85
C10	15	48.22	18.56	4.79
Difference	15	6.77	10.12	2.61

95% CI for mean difference: (1.17, 12.37)
T-test of mean difference = 0 (vs not = 0): T-Value = 2.59 P-Value = 0.021

100 K make-to-stock vs. Extra Injection make-to-stock ORDER COSTS

Paired T-test and CI: C11, C12

Paired T for C11 - C12

	N	Mean	StDev	SE Mean
C11	15	90.612	1.985	0.512
C12	15	90.612	1.985	0.512
Difference	15	0.000000	0.000000	0.000000

95% CI for mean difference: (0.000000, 0.000000)
T-test of mean difference = 0 (vs not = 0): T-Value = * P-Value = *

* NOTE * All values in column are identical.

Increased Demand Variation make-to-order vs. make-to-stock ORDER LEAD TIMES

Paired T-test and CI: C1, C2

Paired T for C1 - C2

	N	Mean	StDev	SE Mean
C1	15	9.918	2.742	0.708
C2	15	2.910	0.480	0.124
Difference	15	7.008	2.876	0.743

95% CI for mean difference: (5.415, 8.600)
T-test of mean difference = 0 (vs not = 0): T-Value = 9.44 P-Value = 0.000

Increased Demand Variation make-to-order vs. make-to-stock ORDER Costs

Paired T-test and CI: C3, C4

Paired T for C3 - C4

	N	Mean	StDev	SE Mean
C3	15	50.36	16.21	4.19
C4	15	90.71	1.99	0.51
Difference	15	-40.36	15.01	3.88

95% CI for mean difference: (-48.67, -32.04)
T-test of mean difference = 0 (vs not = 0): T-Value = -10.41 P-Value = 0.000

100 K make-to-order vs. Increased Demand Variation make-to-order ORDER LEAD TIMES

Paired T-test and CI: C5, C6

Paired T for C5 - C6

	N	Mean	StDev	SE Mean
C5	15	11.13	4.72	1.22
C6	15	9.92	2.74	0.71
Difference	15	1.213	3.188	0.823

95% CI for mean difference: (-0.552, 2.979)
T-test of mean difference = 0 (vs not = 0): T-Value = 1.47 P-Value = 0.163

100 K make-to-order vs. Increased Demand Variation make-to-order ORDER COSTS

Paired T-test and CI: C7, C8

Paired T for C7 - C8

	N	Mean	StDev	SE Mean
C7	15	2.876	0.505	0.130
C8	15	2.910	0.480	0.124
Difference	15	-0.0342	0.2475	0.0639

95% CI for mean difference: (-0.1713, 0.1028)
T-test of mean difference = 0 (vs not = 0): T-Value = -0.54 P-Value = 0.600

100 K make-to-order vs. Increased Demand Variation make-to-order ORDER COSTS

Paired T-test and CI: C9, C10

Paired T for C9 - C10

	N	Mean	StDev	SE Mean
C9	15	54.99	22.64	5.85
C10	15	50.36	16.21	4.19
Difference	15	4.64	10.84	2.80

95% CI for mean difference: (-1.36, 10.64)
T-test of mean difference = 0 (vs not = 0): T-Value = 1.66 P-Value = 0.120

100 K make-to-stock vs. Increased Demand Variation make-to-stock ORDER COSTS

Paired T-test and CI: C11, C12

Paired T for C11 - C12

	N	Mean	StDev	SE Mean
C11	15	90.612	1.985	0.512
C12	15	90.711	1.994	0.515
Difference	15	-0.099	0.684	0.177

95% CI for mean difference: (-0.477, 0.280)
T-test of mean difference = 0 (vs not = 0): T-Value = -0.56 P-Value = 0.585

One Product per Order make-to-order vs. make-to-stock ORDER LEAD TIMES

Paired T-test and CI: C1, C2

Paired T for C1 - C2

	N	Mean	StDev	SE Mean
C1	5	8.98	4.35	1.95
C2	5	2.92	0.60	0.27
Difference	5	6.06	4.81	2.15

95% CI for mean difference: (0.08, 12.04)
T-test of mean difference = 0 (vs not = 0): T-Value = 2.81 P-Value = 0.048
One Product per Order make-to-order vs. make-to-stock ORDER COSTS

Paired T-test and CI: C3, C4

Paired T for C3 - C4

	N	Mean	StDev	SE Mean
C3	5	47.8	24.2	10.8
C4	5	90.4	1.1	0.5
Difference	5	-42.6	23.4	10.5

95% CI for mean difference: (-71.6, -13.6)
T-test of mean difference = 0 (vs not = 0): T-Value = -4.08 P-Value = 0.015

100 K make-to-order vs. One Product per Order make-to-order ORDER LEAD TIMES

Paired T-test and CI: C5, C6

Paired T for C5 - C6

	N	Mean	StDev	SE Mean
C5	5	10.14	5.95	2.66
C6	5	8.98	4.35	1.95
Difference	5	1.16	2.63	1.18

95% CI for mean difference: (-2.11, 4.43)
T-test of mean difference = 0 (vs not = 0): T-Value = 0.99 P-Value = 0.380

100 K make-to-stock vs. One Product per Order make-to-stock ORDER LEAD TIMES

Paired T-test and CI: C7, C8

Paired T for C7 - C8

	N	Mean	StDev	SE Mean
C7	5	2.765	0.402	0.180
C8	5	2.922	0.595	0.266
Difference	5	-0.156	0.542	0.242

95% CI for mean difference: (-0.829, 0.517)
T-test of mean difference = 0 (vs not = 0): T-Value = -0.64 P-Value = 0.555

100 K make-to-order vs. One Product per Order make-to-order ORDER COSTS

Paired T-test and CI: C9, C10

Paired T for C9 - C10

	N	Mean	StDev	SE Mean
C9	5	52.2	31.1	13.9
C10	5	47.8	24.2	10.8
Difference	5	4.45	9.76	4.36

95% CI for mean difference: (-7.67, 16.56)
T-test of mean difference = 0 (vs not = 0): T-Value = 1.02 P-Value = 0.366

100 K make-to-stock vs. One Product per Order make-to-stock ORDER COSTS

Paired T-test and CI: C11, C12

Paired T for C11 - C12

	N	Mean	StDev	SE Mean
C11	5	90.064	1.992	0.891
C12	5	90.402	1.070	0.479
Difference	5	-0.338	1.541	0.689

95% CI for mean difference: (-2.252, 1.576)
T-test of mean difference = 0 (vs not = 0): T-Value = -0.49 P-Value = 0.650

Four Types of Product in Product Family make-to-order vs. make-to-stock ORDER LEAD TIMES

Paired T-test and CI: C1, C2

Paired T for C1 - C2

	N	Mean	StDev	SE Mean
C1	14	11.33	4.83	1.29
C2	14	2.94	0.56	0.15
Difference	14	8.39	4.90	1.31

95% CI for mean difference: (5.56, 11.22)
T-test of mean difference = 0 (vs not = 0): T-Value = 6.41 P-Value = 0.000

Four Types of Product in Product Family make-to-order vs. make-to-stock ORDER COSTS

Paired T-test and CI: C3, C4

Paired T for C3 - C4

	N	Mean	StDev	SE Mean
C3	14	56.47	22.64	6.05
C4	14	91.06	2.12	0.57
Difference	14	-34.58	21.61	5.78

95% CI for mean difference: (-47.06, -22.10)
T-test of mean difference = 0 (vs not = 0): T-Value = -5.99 P-Value = 0.000

100 K make-to-order vs. Four Types of Product in Product Family make-to-order ORDER LEAD TIMES

Paired T-test and CI: C5, C6

Paired T for C5 - C6

	N	Mean	StDev	SE Mean
C5	14	11.33	4.83	1.29
C6	14	11.33	4.83	1.29
Difference	14	0.000000	0.000000	0.000000

95% CI for mean difference: (0.000000, 0.000000)
T-test of mean difference = 0 (vs not = 0): T-Value = * P-Value = *

* NOTE * All values in column are identical.

100 K make-to-stock vs. Four Types of Product in Product Family make-to-stock ORDER COSTS

Paired T-test and CI: C7, C8

Paired T for C7 - C8

	N	Mean	StDev	SE Mean
C7	14	2.855	0.518	0.138
C8	14	2.939	0.561	0.150
Difference	14	-0.084	0.429	0.115

95% CI for mean difference: (-0.332, 0.163)
T-test of mean difference = 0 (vs not = 0): T-Value = -0.74 P-Value = 0.474

100 K make-to-order vs. Four Types of Product in Product Family make-to-order ORDER COSTS

Paired T-test and CI: C9, C10

Paired T for C9 - C10

	N	Mean	StDev	SE Mean
C9	14	56.49	22.72	6.07
C10	14	56.47	22.64	6.05
Difference	14	0.0114	0.1057	0.0282

95% CI for mean difference: (-0.0496, 0.0724)
T-test of mean difference = 0 (vs not = 0): T-Value = 0.40 P-Value = 0.692

100 K make-to-stock vs. Four Types of Product in Product Family make-to-stock ORDER COSTS

Paired T-test and CI: C11, C12

Paired T for C11 - C12

	N	Mean	StDev	SE Mean

	N	Mean	StDev	SE Mean
C11	14	90.812	1.896	0.507
C12	14	91.055	2.116	0.565
Difference	14	-0.243	1.188	0.318

95% CI for mean difference: (-0.929, 0.443)
T-test of mean difference = 0 (vs not = 0): T-Value = -0.76 P-Value = 0.458

Four Types of Product in Product Family One type of Product in order make-to-order vs. make-to-stock ORDER LEAD TIMES

Paired T-test and CI: C1, C2

Paired T for C1 - C2

	N	Mean	StDev	SE Mean
C1	4	9.90	4.44	2.22
C2	4	2.75	0.56	0.28
Difference	4	7.15	4.86	2.43

95% CI for mean difference: (-0.58, 14.87)
T-test of mean difference = 0 (vs not = 0): T-Value = 2.94 P-Value = 0.060

Four Types of Product in Product Family One type of Product in order make-to-order vs. make-to-stock ORDER COSTS

Paired T-test and CI: C3, C4

Paired T for C3 - C4

	N	Mean	StDev	SE Mean
C3	4	53.1	24.4	12.2
C4	4	90.8	0.5	0.2
Difference	4	-37.7	23.9	12.0

95% CI for mean difference: (-75.7, 0.4)
T-test of mean difference = 0 (vs not = 0): T-Value = -3.15 P-Value = 0.051

100 K make-to-order vs. Four Types of Product in Product Family One type of Product in order make-to-order ORDER LEAD TIMES

Paired T-test and CI: C5, C6

Paired T for C5 - C6

	N	Mean	StDev	SE Mean
C5	4	11.55	5.81	2.90

	N	Mean	StDev	SE Mean
C6	4	9.90	4.44	2.22
Difference	4	1.65	5.64	2.82

95% CI for mean difference: (-7.32, 10.63)
T-test of mean difference = 0 (vs not = 0): T-Value = 0.59 P-Value = 0.599

100 K make-to-stock vs. Four Types of Product in Product Family One type of Product in order make-to-stock ORDER LEAD TIMES

Paired T-test and CI: C7, C8

Paired T for C7 - C8

	N	Mean	StDev	SE Mean
C7	4	2.825	0.549	0.274
C8	4	2.750	0.560	0.280
Difference	4	0.075	0.918	0.459

95% CI for mean difference: (-1.386, 1.537)
T-test of mean difference = 0 (vs not = 0): T-Value = 0.16 P-Value = 0.880

100 K make-to-order vs. Four Types of Product in Product Family One type of Product in order make-to-order ORDER COSTS

Paired T-test and CI: C9, C10

Paired T for C9 - C10

	N	Mean	StDev	SE Mean
C9	4	59.0	31.4	15.7
C10	4	53.1	24.4	12.2
Difference	4	5.9	35.1	17.6

95% CI for mean difference: (-50.0, 61.8)
T-test of mean difference = 0 (vs not = 0): T-Value = 0.34 P-Value = 0.759

100 K make-to-stock vs. Four Types of Product in Product Family One type of Product in order make-to-stock ORDER COSTS

Paired T-test and CI: C11, C12

Paired T for C11 - C12

	N	Mean	StDev	SE Mean
C11	4	90.742	1.662	0.831
C12	4	90.790	0.483	0.241
Difference	4	-0.047	1.535	0.767

95% CI for mean difference: (-2.490, 2.395)
T-test of mean difference = 0 (vs not = 0): T-Value = -0.06 P-Value = 0.955

Three Types of Product in Product Family One type of Product in order make-to-order vs. make-to-stock ORDER LEAD TIMES

Paired T-test and CI: C1, C2

Paired T for C1 - C2

	N	Mean	StDev	SE Mean
C1	14	11.42	4.71	1.26
C2	14	2.84	0.54	0.15
Difference	14	8.57	4.84	1.29

95% CI for mean difference: (5.78, 11.37)
T-test of mean difference = 0 (vs not = 0): T-Value = 6.62 P-Value = 0.000

Three Types of Product in Product Family One type of Product in order make-to-order vs. make-to-stock ORDER COSTS

Paired T-test and CI: C3, C4

Paired T for C3 - C4

	N	Mean	StDev	SE Mean
C3	14	56.72	22.31	5.96
C4	14	90.76	1.74	0.46
Difference	14	-34.04	21.47	5.74

95% CI for mean difference: (-46.44, -21.64)
T-test of mean difference = 0 (vs not = 0): T-Value = -5.93 P-Value = 0.000

100 K make-to-order vs. Three Types of Product in Product Family One type of Product in order make-to-order ORDER LEAD TIMES

Paired T-test and CI: C5, C1

Paired T for C5 - C1

	N	Mean	StDev	SE Mean
C5	14	11.40	4.73	1.26
C1	14	11.42	4.71	1.26
Difference	14	-0.0142	0.0533	0.0142

95% CI for mean difference: (-0.0450, 0.0165)
T-test of mean difference = 0 (vs not = 0): T-Value = -1.00 P-Value = 0.336

100 K make-to-stock vs. Three Types of Product in Product Family One type of Product in order make-to-stock ORDER LEAD TIMES

Paired T-test and CI: C6, C2

Paired T for C6 - C2

	N	Mean	StDev	SE Mean
C6	13	2.861	0.538	0.149
C2	13	2.801	0.545	0.151
Difference	13	0.0596	0.2798	0.0776

95% CI for mean difference: (-0.1095, 0.2287)
T-test of mean difference = 0 (vs not = 0): T-Value = 0.77 P-Value = 0.457

100 K make-to-order vs. Three Types of Product in Product Family One type of Product in order make-to-order ORDER COSTS

Paired T-test and CI: C1, C2

Paired T for C1 - C2

	N	Mean	StDev	SE Mean
C1	13	58.78	21.89	6.07
C2	13	58.77	21.80	6.05
Difference	13	0.0154	0.1089	0.0302

95% CI for mean difference: (-0.0504, 0.0812)
T-test of mean difference = 0 (vs not = 0): T-Value = 0.51 P-Value = 0.620

100 K make-to-stock vs. Three Types of Product in Product Family One type of Product in order make-to-stock ORDER COSTS

Paired T-test and CI: C3, C4

Paired T for C3 - C4

	N	Mean	StDev	SE Mean
C3	13	90.943	1.906	0.529
C4	13	90.772	1.807	0.501
Difference	13	0.172	0.806	0.223

95% CI for mean difference: (-0.315, 0.658)
T-test of mean difference = 0 (vs not = 0): T-Value = 0.77 P-Value = 0.457

Appendix D

Remaining Tables of Full Factorial Design

The Lead Time Values for the 80 K Demand, Three Types of Product in Product Family Model with the Make-to-order and Make-to-stock

80K Demand make-to-order 3 Types of Screwdrivers Order Lead Times	**Hr.**	**80K Demand make-to-stock 3 Types of Screwdrivers Order Lead Times**	**Hr.**
Order Type 3 (p3)	4.3564	Order Type 3 (p3)	2.6393
Order Type 4 (p4)	9.303	Order Type 4 (p4)	1.7884
Order Type 5 (p5)	10.567	Order Type 5 (p5)	1.9661
Order Type 6 (p1-p2)	5.5647	Order Type 6 (p1-p2)	3.7962
Order Type 7 (p1-p3)	4.383	Order Type 7 (p1-p3)	2.5512
Order Type 8 (p1-p4)	8.6307	Order Type 8 (p1-p4)	2.2888
Order Type 9 (p1-p5)	6.638	Order Type 9 (p1-p5)	2.1378
Order Type 10 (p2-p3)	4.1191	Order Type 10 (p2-p3)	2.5399
Order Type 11 (p2-p4)	9.3873	Order Type 11 (p2-p4)	2.6277
Order Type 12 (p2-p5)	7.3778	Order Type 12 (p2-p5)	2.8812
Order Type 13 (p3-p4)	6.3718	Order Type 13 (p3-p4)	3.1282
Order Type 14 (p3-p5)	7.4556	Order Type 14 (p3-p5)	3.185
Order Type 15 (p4-p5)	9.0679	Order Type 15 (p4-p5)	1.7743
80K Demand make-to-order 3 Types of Screwdrivers Product Lead Times		**80K Demand make-to-stock 3 Types of Screwdrivers Product Lead Times**	
P3	2.8752	P3	1.5241
P4	5.5402	P4	1.5953
P5	5.6733	P5	1.8517

The Cost Values for the 80 K Demand, Three Types of Product in Product Family Model with the Make-to-order and Make-to-stock

80K Demand make-to-order 3 Types Screwdrivers Order Cost	$ Fixed Cost	$ WIP Inven. Cost	$ Total Cost	80K Demand make-to-stock 3 Types Screwdrivers Order Cost	$ Fixed Cost	$ WIP Inven. Cost	$ Ave. Inven. Holding Cost	$ Total Cost
Order Type 1 (p1)	-	-	-	Order Type 1 (p1)	-	-	-	-
Order Type 2 (p2)	-	-	-	Order Type 2 (p2)	-	-	-	-
Order Type 3 (p3)	14.8	12.89	27.69	Order Type 3 (p3)	13.44	7.09	69.15	89.68
Order Type 4 (p4)	18.8	34.98	53.78	Order Type 4 (p4)	14.08	5.04	69.15	88.27
Order Type 5 (p5)	20.8	43.96	64.76	Order Type 5 (p5)	15.78	6.21	69.15	91.14
Order Type 6 (p1-p2)	13.5	15.02	28.52	Order Type 6 (p1-p2)	12.12	9.20	69.15	90.47
Order Type 7 (p1-p3)	13.8	12.10	25.90	Order Type 7 (p1-p3)	12.43	6.34	69.15	87.92
Order Type 8 (p1-p4)	15.8	27.27	43.07	Order Type 8 (p1-p4)	12.75	5.84	69.15	87.74
Order Type 9 (p1-p5)	20.8	27.61	48.41	Order Type 9 (p1-p5)	15.78	6.75	69.15	91.68
Order Type 10 (p2-p3)	14.5	11.95	26.45	Order Type 10 (p2-p3)	13.13	6.67	69.15	88.95
Order Type 11 (p2-p4)	16.5	30.98	47.48	Order Type 11 (p2-p4)	13.45	7.07	69.15	89.67
Order Type 12 (p2-p5)	17.5	25.82	43.32	Order Type 12 (p2-p5)	14.3	8.24	69.15	91.69
Order Type 13 (p3-p4)	16.8	21.41	38.21	Order Type 13 (p3-p4)	13.76	8.61	69.15	91.52
Order Type 14 (p3-p5)	17.8	26.54	44.34	Order Type 14 (p3-p5)	14.61	9.31	69.15	93.07
Order Type 15 (p4-p5)	19.8	35.91	55.71	Order Type 15 (p4-p5)	14.93	5.30	69.15	89.38
80K Demand make-to-order 3 Types Screwdrivers Product Cost				**80K Demand make-to-stock 3 Types Screwdrivers Product Cost**				
P1	-	-	-	P1	-	-	-	-
P2	-	-	-	P2	-	-	-	-
P3	14.75	8.48	23.23	P3	13.44	4.10	69.15	86.69
P4	18.84	20.88	39.72	P4	14.08	4.49	69.15	87.72
P5	20.84	23.65	44.49	P5	15.78	5.84	69.15	90.77

The Lead Time Values for the 80 K Demand, Four Types of Product in Product Family Model with the Make-to-order and Make-to-stock

80K Demand make-to-order 4 Types of Screwdrivers Order Lead Times	**Hr.**	**80K Demand make-to-stock 4 Types of Screwdrivers Order Lead Times**	**Hr.**
Order Type 1 (p1)	-	Order Type 1 (p1)	-
Order Type 2 (p2)	3.217	Order Type 2 (p2)	2.1996
Order Type 3 (p3)	4.3564	Order Type 3 (p3)	3.4053
Order Type 4 (p4)	9.303	Order Type 4 (p4)	1.6293
Order Type 5 (p5)	10.567	Order Type 5 (p5)	2.2108
Order Type 6 (p1-p2)	5.5647	Order Type 6 (p1-p2)	3.291
Order Type 7 (p1-p3)	4.383	Order Type 7 (p1-p3)	2.84
Order Type 8 (p1-p4)	8.6307	Order Type 8 (p1-p4)	2.0434
Order Type 9 (p1-p5)	6.638	Order Type 9 (p1-p5)	2.1969
Order Type 10 (p2-p3)	4.1191	Order Type 10 (p2-p3)	2.3727
Order Type 11 (p2-p4)	9.3873	Order Type 11 (p2-p4)	2.8143
Order Type 12 (p2-p5)	7.3778	Order Type 12 (p2-p5)	1.8095
Order Type 13 (p3-p4)	6.3718	Order Type 13 (p3-p4)	3.0694
Order Type 14 (p3-p5)	7.4556	Order Type 14 (p3-p5)	2.6457
Order Type 15 (p4-p5)	9.0679	Order Type 15 (p4-p5)	1.8942
80K Demand make-to-order 4 Types of Screwdrivers Product Lead Times		**80K Demand make-to-stock 4 Types of Screwdrivers Product Lead Times**	
P1	-	P1	-
P2	2.9787	P2	1.189
P3	2.6531	P3	1.81
P4	5.5402	P4	1.5358
P5	5.6733	P5	1.6836

The Cost Values for the 80 K Demand, Four Types of Product in Product Family Model with the Make-to-order and Make-to-stock

80K Demand make-to-order 4 Types Screwdrivers Order Cost	**$ Fixed Cost**	**$ WIP Inven. Cost**	**$ Total Cost**	**80K Demand make-to-stock 4 Types Screwdrivers Order Cost**	**$ Fixed Cost**	**$ WIP Inven. Cost**	**$ Ave. Inven. Holding Cost**	**$ Total Cost**
Order Type 1 (p1)	-	-	-	Order Type 1 (p1)	-	-	-	-
Order Type 2 (p2)	-	9.14	23.34	Order Type 2 (p2)	12.82	5.64	69.15	87.61
Order Type 3 (p3)	14.8	12.89	27.69	Order Type 3 (p3)	13.44	9.15	69.15	91.74
Order Type 4 (p4)	18.8	34.98	53.78	Order Type 4 (p4)	14.08	4.59	69.15	87.82
Order Type 5 (p5)	20.8	43.96	64.76	Order Type 5 (p5)	15.78	6.98	69.15	91.91
Order Type 6 (p1-p2)	13.5	15.02	28.52	Order Type 6 (p1-p2)	12.12	7.98	69.15	89.25
Order Type 7 (p1-p3)	13.8	12.10	25.90	Order Type 7 (p1-p3)	12.43	7.06	69.15	88.64
Order Type 8 (p1-p4)	15.8	27.27	43.07	Order Type 8 (p1-p4)	12.75	5.21	69.15	87.11
Order Type 9 (p1-p5)	20.8	27.61	48.41	Order Type 9 (p1-p5)	15.78	6.93	69.15	91.86
Order Type 10 (p2-p3)	14.5	11.95	26.45	Order Type 10 (p2-p3)	13.13	6.23	69.15	88.51
Order Type 11 (p2-p4)	16.5	30.98	47.48	Order Type 11 (p2-p4)	13.45	7.57	69.15	90.17
Order Type 12 (p2-p5)	17.5	25.82	43.32	Order Type 12 (p2-p5)	14.3	5.18	69.15	88.63
Order Type 13 (p3-p4)	16.8	21.41	38.21	Order Type 13 (p3-p4)	13.76	8.45	69.15	91.36
Order Type 14 (p3-p5)	17.8	26.54	44.34	Order Type 14 (p3-p5)	14.61	7.73	69.15	91.49
Order Type 15 (p4-p5)	19.8	35.91	55.71	Order Type 15 (p4-p5)	14.93	5.66	69.15	89.74
80K Demand make-to-order 4 Types Screwdrivers Product Cost				**80K Demand make-to-stock 4 Types Screwdrivers Product Cost**				
P1	-	-	-	P1	-	-	-	-
P2	14.17	8.44	22.61	P2	12.82	3.049	69.15	85.0
P3	14.75	7.83	22.58	P3	13.44	4.865	69.15	87.5
P4	18.84	20.88	39.72	P4	14.08	4.325	69.15	87.6
P5	20.84	23.65	44.49	P5	15.78	5.313	69.15	90.2

The Lead Time Values for the 120 K Demand Three Types of Product in Product Family Model with the Make-to-order and Make-to-stock

120K Demand make-to-order 3 Types of Screwdrivers Order Lead Times	Hr.	120K Demand make-to-stock 3 Types of Screwdrivers Order Lead Times	Hr.
Order Type 1 (p1)	-	Order Type 1 (p1)	-
Order Type 2 (p2)	-	Order Type 2 (p2)	-
Order Type 3 (p3)	12.22	Order Type 3 (p3)	3.90
Order Type 4 (p4)	15.76	Order Type 4 (p4)	2.54
Order Type 5 (p5)	16.38	Order Type 5 (p5)	3.03
Order Type 6 (p1-p2)	7.17	Order Type 6 (p1-p2)	4.03
Order Type 7 (p1-p3)	14.47	Order Type 7 (p1-p3)	4.30
Order Type 8 (p1-p4)	8.72	Order Type 8 (p1-p4)	3.18
Order Type 9 (p1-p5)	8.69	Order Type 9 (p1-p5)	3.16
Order Type 10 (p2-p3)	7.67	Order Type 10 (p2-p3)	4.64
Order Type 11 (p2-p4)	14.47	Order Type 11 (p2-p4)	5.81
Order Type 12 (p2-p5)	9.62	Order Type 12 (p2-p5)	2.55
Order Type 13 (p3-p4)	14.90	Order Type 13 (p3-p4)	4.66
Order Type 14 (p3-p5)	9.54	Order Type 14 (p3-p5)	2.74
Order Type 15 (p4-p5)	15.94	Order Type 15 (p4-p5)	2.71
120K Demand make-to-order 3 Types of Screwdrivers Product Lead Times		**120K Demand make-to-stock 3 Types of Screwdrivers Product Lead Times**	
P1	-	P1	-
P2	-	P2	-
P3	5.28	P3	2.18
P4	9.48	P4	2.58
P5	8.70	P5	2.15

The Cost Values for the 120 K Demand Three Types of Product in Product Family Model with the Make-to-order and Make-to-stock

120K Demand make-to-order 3 Types Screwdrivers Order Cost	**$ Fixed Cost**	**$ WIP Inven. Cost**	**$ Total Cost**	**120K Demand make-to-stock 3 Types Screwdrivers Order Cost**	**$ Fixed Cost**	**$ WIP Inven. Cost**	**$ Ave. Inven. Holding Cost**	**$ Total Cost**
Order Type 1 (p1)	-	-	-	Order Type 1 (p1)	-	-	-	-
Order Type 2 (p2)	-	-	-	Order Type 2 (p2)	12.82	-	-	-
Order Type 3 (p3)	14.8	36.17	50.97	Order Type 3 (p3)	13.44	10.49	69.15	93.08
Order Type 4 (p4)	18.8	59.26	78.06	Order Type 4 (p4)	14.08	7.15	69.15	90.38
Order Type 5 (p5)	20.8	68.14	88.94	Order Type 5 (p5)	15.78	9.55	69.15	94.48
Order Type 6 (p1-p2)	13.5	19.35	32.85	Order Type 6 (p1-p2)	12.12	9.76	69.15	91.03
Order Type 7 (p1-p3)	13.8	39.95	53.75	Order Type 7 (p1-p3)	12.43	10.69	69.15	92.27
Order Type 8 (p1-p4)	15.8	27.55	43.35	Order Type 8 (p1-p4)	12.75	8.10	69.15	90.00
Order Type 9 (p1-p5)	20.8	36.17	56.97	Order Type 9 (p1-p5)	15.78	9.98	69.15	94.91
Order Type 10 (p2-p3)	14.5	22.25	36.75	Order Type 10 (p2-p3)	13.13	12.19	69.15	94.47
Order Type 11 (p2-p4)	16.5	47.76	64.26	Order Type 11 (p2-p4)	13.45	15.63	69.15	98.23
Order Type 12 (p2-p5)	17.5	33.68	51.18	Order Type 12 (p2-p5)	14.3	7.30	69.15	90.75
Order Type 13 (p3-p4)	16.8	50.08	66.88	Order Type 13 (p3-p4)	13.76	12.84	69.15	95.75
Order Type 14 (p3-p5)	17.8	33.95	51.75	Order Type 14 (p3-p5)	14.61	8.00	69.15	91.76
Order Type 15 (p4-p5)	19.8	63.13	82.93	Order Type 15 (p4-p5)	14.93	8.08	69.15	92.16
120K Demand make-to-order 3 Types Screwdrivers Product Cost				**120K Demand make-to-stock 3 Types Screwdrivers Product Cost**				
P1	-	-	-	P1	-	-	-	-
P2	-	-	-	P2	12.82	-	-	-
P3	14.75	15.57	30.32	P3	13.44	5.86	69.15	88.45
P4	18.84	35.71	54.55	P4	14.08	7.26	69.15	90.49
P5	20.84	36.26	57.10	P5	15.78	6.79	69.15	91.72

The Lead Time Values for the 120 K Demand Four Types of Product in Product Family Model with the Make-to-order and Make-to-stock

120K Demand make-to-order 4 Types of Screwdrivers Order Lead Times	Hr.	120K Demand make-to-stock 4 Types of Screwdrivers Order Lead Times	Hr.
Order Type 1 (p1)	-	Order Type 1 (p1)	-
Order Type 2 (p2)	4.498	Order Type 2 (p2)	3.2828
Order Type 3 (p3)	12.221	Order Type 3 (p3)	3.1069
Order Type 4 (p4)	15.761	Order Type 4 (p4)	2.9384
Order Type 5 (p5)	16.38	Order Type 5 (p5)	3.1158
Order Type 6 (p1-p2)	7.167	Order Type 6 (p1-p2)	4.2774
Order Type 7 (p1-p3)	7.5275	Order Type 7 (p1-p3)	4.1502
Order Type 8 (p1-p4)	8.718	Order Type 8 (p1-p4)	4.4277
Order Type 9 (p1-p5)	8.6938	Order Type 9 (p1-p5)	4.0563
Order Type 10 (p2-p3)	7.6726	Order Type 10 (p2-p3)	4.1118
Order Type 11 (p2-p4)	14.474	Order Type 11 (p2-p4)	4.5418
Order Type 12 (p2-p5)	9.6226	Order Type 12 (p2-p5)	2.7974
Order Type 13 (p3-p4)	14.904	Order Type 13 (p3-p4)	5.7565
Order Type 14 (p3-p5)	9.5357	Order Type 14 (p3-p5)	3.7697
Order Type 15 (p4-p5)	15.941	Order Type 15 (p4-p5)	2.3017
120K Demand make-to-order 4 Types of Screwdrivers Product Lead Times		**120K Demand make-to-stock 4 Types of Screwdrivers Product Lead Times**	
P1	-	P1	-
P2	4.2345	P2	1.9243
P3	7.5911	P3	2.5535
P4	9.477	P4	2.6828
P5	8.6993	P5	2.4947

The Cost Values for the Four Types of Product in Product Family Model with the Make-to-order and Make-to-stock

120K Demand make-to-order 4 Types Screwdrivers Order Cost	**$ Fixed Cost**	**$ WIP Inven. Cost**	**$ Total Cost**	**120K Demand make-to-stock 4 Types Screwdrivers Order Cost**	**$ Fixed Cost**	**$ WIP Inven. Cost**	**$ Ave. Inven. Holding Cost**	**$ Total Cost**
Order Type 1 (p1)	-	-	-	Order Type 1 (p1)	-	-	-	-
Order Type 2 (p2)	-	9.14	23.34	Order Type 2 (p2)	12.82	8.42	69.15	90.39
Order Type 3 (p3)	14.8	12.89	27.69	Order Type 3 (p3)	13.44	8.35	69.15	90.94
Order Type 4 (p4)	18.8	34.98	53.78	Order Type 4 (p4)	14.08	8.27	69.15	91.50
Order Type 5 (p5)	20.8	43.96	64.76	Order Type 5 (p5)	15.78	9.83	69.15	94.76
Order Type 6 (p1-p2)	13.5	15.02	28.52	Order Type 6 (p1-p2)	12.12	10.37	69.15	91.64
Order Type 7 (p1-p3)	13.8	12.10	25.90	Order Type 7 (p1-p3)	12.43	10.32	69.15	91.90
Order Type 8 (p1-p4)	15.8	27.27	43.07	Order Type 8 (p1-p4)	12.75	11.29	69.15	93.19
Order Type 9 (p1-p5)	20.8	27.61	48.41	Order Type 9 (p1-p5)	15.78	12.80	69.15	97.73
Order Type 10 (p2-p3)	14.5	11.95	26.45	Order Type 10 (p2-p3)	13.13	10.80	69.15	93.08
Order Type 11 (p2-p4)	16.5	30.98	47.48	Order Type 11 (p2-p4)	13.45	12.22	69.15	94.82
Order Type 12 (p2-p5)	17.5	25.82	43.32	Order Type 12 (p2-p5)	14.3	8.00	69.15	91.45
Order Type 13 (p3-p4)	16.8	21.41	38.21	Order Type 13 (p3-p4)	13.76	15.84	69.15	98.75
Order Type 14 (p3-p5)	17.8	26.54	44.34	Order Type 14 (p3-p5)	14.61	11.02	69.15	94.78
Order Type 15 (p4-p5)	19.8	35.91	55.71	Order Type 15 (p4-p5)	14.93	6.87	69.15	90.95
120K Demand make-to-order 4 Types Screwdrivers Product Cost				**120K Demand make-to-stock 4 Types Screwdrivers Product Cost**				
P1	-	-	-	P1	-	-	-	-
P2	14.17	12.77	26.97	P2	12.82	4.93	69.15	86.904
P3	14.75	36.17	50.97	P3	13.44	6.86	69.15	89.454
P4	18.84	59.26	78.06	P4	14.08	7.55	69.15	90.785
P5	20.84	68.14	88.94	P5	15.78	7.87	69.15	92.803

Appendix E

ANOVA Tables

ANOVA: C4 versus C1, C2, C3

Factor	Type	Levels	Values
C1	fixed	3	80, 100, 120
C2	fixed	2	1, 2
C3	fixed	3	3, 4, 5

Analysis of Variance for C4

Source	DF	SS	MS	F	P
C1	2	48.020	24.010	5.63	0.019
C2	1	265.421	265.421	62.19	0.000
C3	2	3.811	1.905	0.45	0.650
Error	12	51.217	4.268		
Total	17	368.469			

S = 2.06594   R-Sq = 86.10%   R-Sq(adj) = 80.31%

ANOVA: C5 versus C1, C2, C3

Factor	Type	Levels	Values
C1	fixed	3	80, 100, 120
C2	fixed	2	1, 2
C3	fixed	3	3, 4, 5

Analysis of Variance for C5

Source	DF	SS	MS	F	P
C1	2	378.5	189.3	2.83	0.099
C2	1	6848.0	6848.0	102.32	0.000
C3	2	90.2	45.1	0.67	0.528
Error	12	803.1	66.9		
Total	17	8119.8			

S = 8.18088   R-Sq = 90.11%   R-Sq(adj) = 85.99%

ANOVA: C4, C5 versus C1, C2, C3

MANOVA for C1
s = 2 m = -0.5 n = 4.5

Criterion	Test Statistic	F	DF Num	DF Denom	P
Wilks'	0.39597	3.240	4	22	0.031
Lawley-Hotelling	1.40745	3.519	4	20	0.025
Pillai's	0.65075	2.894	4	24	0.044
Roy's	1.31791				

MANOVA for C2
s = 1 m = 0.0 n = 4.5

Criterion	Test Statistic	F	DF Num	DF Denom	P
Wilks'	0.00551	992.760	2	11	0.000
Lawley-Hotelling	180.50175	992.760	2	11	0.000
Pillai's	0.99449	992.760	2	11	0.000
Roy's	180.50175				

MANOVA for C3
s = 2 m = -0.5 n = 4.5

Criterion	Test Statistic	F	DF Num	DF Denom	P
Wilks'	0.80173	0.643	4	22	0.638
Lawley-Hotelling	0.23652	0.591	4	20	0.673
Pillai's	0.20690	0.692	4	24	0.605
Roy's	0.17495				

ANOVA: C5 versus C1, C2, C3

Factor	Type	Levels	Values
C1	fixed	3	80, 100, 120
C2	fixed	2	1, 2
C3	fixed	3	3, 4, 5

Analysis of Variance for C5

Source	DF	SS	MS	F	P
C1	2	378.5	189.3	2.83	0.099
C2	1	6848.0	6848.0	102.32	0.000
C3	2	90.2	45.1	0.67	0.528

```
Error   12   803.1    66.9
Total   17  8119.8


S = 8.18088   R-Sq = 90.11%   R-Sq(adj) = 85.99%
```

General Linear Model: C4 versus C1, C2, C3

```
Factor  Type   Levels  Values
C1      fixed       3  80, 100, 120
C2      fixed       2  1, 2
C3      fixed       3  3, 4, 5


Analysis of Variance for C4, using Adjusted SS for Tests

Source  DF   Seq SS   Adj SS   Adj MS      F      P
C1       2   48.020   48.020   24.010   8.61  0.036
C2       1  265.421  265.421  265.421  95.14  0.001
C3       2    3.811    3.811    1.905   0.68  0.556
C1*C2    2   24.732   24.732   12.366   4.43  0.097
C1*C3    4   10.848   10.848    2.712   0.97  0.511
C2*C3    2    4.478    4.478    2.239   0.80  0.509
Error    4   11.159   11.159    2.790
Total   17  368.469


S = 1.67027   R-Sq = 96.97%   R-Sq(adj) = 87.13%
```

Main Effects Plot for C4

Interaction Plot for C4

General Linear Model: C5 versus C1, C2, C3

```
Factor  Type   Levels  Values
C1      fixed       3  80, 100, 120
C2      fixed       2  1, 2
C3      fixed       3  3, 4, 5


Analysis of Variance for C5, using Adjusted SS for Tests

Source  DF   Seq SS   Adj SS   Adj MS      F      P
C1       2   378.51   378.51   189.25    3.11  0.153
```

C2	1	6848.01	6848.01	6848.01	112.68	0.000
C3	2	90.16	90.16	45.08	0.74	0.532
C1*C2	2	215.66	215.66	107.83	1.77	0.281
C1*C3	4	237.59	237.59	59.40	0.98	0.509
C2*C3	2	106.77	106.77	53.38	0.88	0.483
Error	4	243.10	243.10	60.78		
Total	17	8119.80				

S = 7.79591   R-Sq = 97.01%   R-Sq(adj) = 87.28%

Main Effects Plot for C5

Interaction Plot for C5

ANOVA: C4, C5 versus C1, C2, C3

MANOVA for C1
s = 2   m = -0.5   n = 0.5

Criterion	Test Statistic	F	DF Num	DF Denom	P
Wilks'	0.07841	3.857	4	6	0.069
Lawley-Hotelling	9.49164	4.746	4	4	0.080
Pillai's	1.09892	2.439	4	8	0.132
Roy's	9.24707				

MANOVA for C2
s = 1   m = 0.0   n = 0.5

Criterion	Test Statistic	F	DF Num	DF Denom	P
Wilks'	0.00113	1324.860	2	3	0.000
Lawley-Hotelling	883.23979	1324.860	2	3	0.000
Pillai's	0.99887	1324.860	2	3	0.000
Roy's	883.23979				

MANOVA for C3
s = 2   m = -0.5   n = 0.5

Criterion	Test Statistic	F	DF Num	DF Denom	P
Wilks'	0.46456	0.701	4	6	0.619
Lawley-Hotelling	0.94611	0.473	4	4	0.757

Criterion	Test Statistic	F	DF Num	DF Denom	P
Pillai's	0.63136	0.923	4	8	0.496
Roy's	0.60461				

MANOVA for C1*C2
s = 2   m = -0.5   n = 0.5

Criterion	Test Statistic	F	DF Num	DF Denom	P
Wilks'	0.14368	2.457	4	6	0.156
Lawley-Hotelling	4.83566	2.418	4	4	0.207
Pillai's	1.01786	2.073	4	8	0.177
Roy's	4.59076				

MANOVA for C1*C3
s = 2   m = 0.5   n = 0.5

Criterion	Test Statistic	F	DF Num	DF Denom	P
Wilks'	0.24720	0.758	8	6	0.651
Lawley-Hotelling	2.02342	0.506	8	4	0.809
Pillai's	1.00543	1.011	8	8	0.494
Roy's	1.05176				

MANOVA for C2*C3
s = 2   m = -0.5   n = 0.5

Criterion	Test Statistic	F	DF Num	DF Denom	P
Wilks'	0.47944	0.666	4	6	0.638
Lawley-Hotelling	0.89111	0.446	4	4	0.774
Pillai's	0.61388	0.886	4	8	0.514
Roy's	0.50781				