The Pennsylvania State University

The Graduate School

College of Engineering

A SIMULATION OPTIMIZATION TECHNIQUE FOR OPTIMAL RESOURCE ALLOCATION DURING JOB SHOP RAMP-UP

A Thesis in

Industrial Engineering and Operations Research

by

Nicholas A. Kluthe

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

December 2011

The thesis of Nicholas A. Kluthe was reviewed and approved* by the following:

Deborah J. Medeiros
Associate Professor of Industrial Engineering
Thesis Advisor

Mark T. Traband
Affiliate Faculty Member of Industrial Engineering

Jack C. Hayya
Professor Emeritus of Supply Chain and Information Systems

Paul Griffin
Professor of Industrial Engineering
Head of the Department of Industrial Engineering

*Signatures are on file in the Graduate School

Abstract

Recent trends in globalization have led to an increase in competition between manufacturers across many industries that are producing new products with shorter life cycles. The success of these manufacturers is largely due to not only to how quickly they can get these products to market but also how quickly they can produce these products at full volume. This has led to a need for efficient production ramp-up techniques that can determine optimal resource allocation plans for a manufacturing system. This research seeks to develop a method which determines the optimal machine allocation strategy during the ramp-up in a job shop environment. The method uses optimization and simulation techniques to arrive at a solution which is both optimal and robust. The technique is then applied to a case study to demonstrate its use and effectiveness in determining a solution. Such a solution would provide a manufacturing system's decision makers with a production ramp-up strategy that would minimize cost and allow the system to produce the throughput required by its end customers.

Table of Contents

List of Figures

List of Tables

Acknowledgements

This thesis wouldn’t have been possible without the guidance and assistance of several experienced advisors and researchers in the field. First of all, I would like to thank Dr. Deborah Medeiros for her patience and guidance throughout the process of writing this thesis. Her knowledge and expertise have helped shape the work and further increase my interest in the research area. As a professor and an advisor, she has taught me and countless colleagues many useful skills and lessons that will be utilized throughout our careers. Her passion for teaching and research make her an invaluable resource for the Industrial Engineering Department at Penn State. I would also like to acknowledge Dr. Jack Hayya for evaluating the work. Additionally, Dr. Paul Griffin has served an important role as an advisor during my time at Penn State and has given me guidance which will help to shape my career as an industrial engineer.

I would also like to thank the support team that I had at the Applied Research Lab. Dr. Dan Finke, Chris Ligetti and Dr. Mark Traband generously provided their time and guidance on a daily basis to help develop this thesis. Without their numerous years of experience in manufacturing and operations research, this thesis would not have been as comprehensive and would have been much more difficult to compose. They have made many significant contributions to research in several areas and will undoubtedly be successful in their future endeavors. I would also like to thank Lindo Systems Inc. and Simio LLC for providing me with educational licenses for their well-designed software packages that were utilized in this research.

Finally, I would like to thank my parents for the endless support and love they have given me throughout all my life. They have encouraged me in academics, athletics, my career and all aspects of my life. I can only hope to achieve the same success that they have had. They have been the best examples of what I hope to be in the future, and they have shaped who I am today.

Chapter 1

Introduction

1.1 Background Information

The 21st century has brought with it increased competition between companies both domestically and abroad. New global markets and more demanding consumer behavior further add to the challenges of manufacturing a variety of products in large quantities. More products are constantly being introduced, and they are experiencing shorter life cycles. This is especially true in electronics and telecommunications industries where products are quickly replaced and made obsolete (Terwiesch and Bohn, 2001). The transition between new products and manufacturing processes can be very costly, especially in process learning and new machine acquisition. This trend in manufacturing has led many companies to explore how to best progress through the design and manufacturing processes.

The term, "ramp-up," refers to the time period between when a product is first introduced to the market and when it reaches full volume production. The ramp-up process occurs every time a new product is introduced in a manufacturing system. It is a critical time in the product's life cycle as well since the product's demand and price are often highest when it is first released (Haller et al., 2003). Hopp and Spearman (2008) have stated that on average "products that come in on budget but 6 months behind schedule sacrifice 33% of profits over 5 years. On-time projects which are 50% over budget lost 4% of profits." At the same time, the production rate of the product is at its lowest point as it is at the beginning of its learning curve. Unfamiliar processes, lower yields, new machines requiring testing and maintenance and high process setup times all contribute to this low initial production rate (Terwiesch and Bohn, 2001). Despite these setbacks, companies that can quickly get their product to market and produce it in large volumes will achieve considerable competitive advantage in the marketplace.

This research will explore production ramp-up in order to determine an optimal method for machine allocation in a job shop manufacturing environment. A job shop provides a strong manufacturing system for experimentation as it is made up of many independent workstations which can have different numbers of machines depending on their utilization. It is a less predictable type of ramp-up than a continuous line manufacturing ramp-up because it is comprised of multiple workstations that act independently and produce multiple product lines in varying amounts. Also, very little research has been conducted in the area of production ramp-up in a job shop environment.

Optimization and simulation will be used to solve the production ramp-up problem and find a near-optimal solution that satisfies user-defined constraints. An integer program first quickly determines the optimal solution in a deterministic system. Its solution is then tested in a simulation model that tests the solution in a stochastic environment where product routings and variability are introduced. The results from this simulation are then factored into the optimization model in order to move toward an optimal and robust solution. This process continues until that solution is determined. Marginal changes to the solution are also able to be made by the decision makers until the performance goals are met.

1.2 Organization of this Research

Following a thorough literature review of past and present production ramp-up research in Chapter 2, a description of the developed algorithm and methodology will be given. This description, available in Chapter 3, will begin by comprehensively describing the problem to be examined as well as the assumptions made in the models that solve the problem. The solution algorithm describes the process that the decision makers follow in order to find a ramp-up solution for the job-shop. It also sets the performance goals of the user that the system aims to fulfill. A detailed look at the optimization and simulation procedures and components used in the algorithm are provided in Sections 3.4 and 3.5.

Chapter 4 will focus on experimentation and analysis of the production ramp-up system. A case study will be explored in order to demonstrate the system's functionality and will look at some of the advantages that it provides over other techniques. The individual data set that is used is explained, and sensitivity analysis is eventually performed to show the effect of changes to the input data. The solution procedure and each iteration of the process is described and analyzed. The results of this case study are then generalized to apply to other job shop systems and other manufacturing environments. Lastly, Chapter 5 will summarize the research and present the conclusions and some possible areas of future research to further advance the field.

Chapter 2

Literature Review

2.1 Overview

This literature review will begin by exploring what production ramp-up means in industry according to many well known researchers in the field. Ramp-up problems will be examined from all different types of manufacturing systems with a focus to how similar problems have been approached in a job shop environment. The literature review will look at how the term has been perceived throughout the past and what changes new technology and research have made to ramp-up methodologies. The review will also look at many authors' different theories and methodologies regarding production ramp-up and how these are applicable to different manufacturing environments. Each different manufacturing scenario requires unique ramp-up strategies. For this reason, there is a wide array of previously practiced methods, each distinct and inventive in its own way.

The purpose of looking into production ramp-up techniques, as is done in this review, is to provide a technique to assist in making critical business decisions. Such a technique could be implemented into a decision support tool or an easy-to-use system for an end user. This idea of a decision support tool or system has been around for decades and has expanded into many different classifications (e.g., Keen 1980a, 1980b; Cox and Adams, 1980; Power, 2007, 2011). Understanding the purpose and ways that decision support systems are used is important to developing a methodology that could drive such a tool that is both flexible enough to work in different situations but is also precise enough to provide a sufficient solution in a given scenario.

The production ramp-up system used to provide solutions for the ramp-up problem in this report has a two-fold solution procedure. First, the problem is optimized for a given level of production, and then the solution is tested using simulation to ensure that the solution is feasible and robust. There are many different optimization techniques that have been used and many different software packages that are now available to help users write and solve mathematical programs of all different types. Mathematical programming is one of the more common techniques of optimization, but researchers have solved resource allocation problems using many different analytical and heuristic approaches (e.g., Fox, 1966; Dallery and Frein, 1988; Dallery and Stecke, 1990; Boxma et al., 1990; Frenk et al., 1994; Terweish and Xu, 2004; Huang et al., 2006).

The second element to the decision support tool is the use of simulation. Many researchers and engineers use simulation in this area of manufacturing as increased customization is available, and it allows systems to be easily manipulated (e.g., Ziegler, 1984, 1990; Ziegler and Praehofer, 2000; Couretas et al, 1998, 1999, 2001). New technologies and software packages allow for more accurate forecasting and resource planning, creating many options and allowing managers to test different situations with several performance measures. Further research has expanded to optimize these simulations and the ramp-up operation that they model, creating new challenges and possibilities (e.g., Harmonosky et al., 1999; Eklin et al., 2009).

Mathematical programming and simulation have been used together previously, but many researchers are hesitant because of the different capabilities of the two methods. Mathematical programming provides a single optimal solution, while simulation introduces variability and tests parameters under these conditions. Many decision makers using simulation to make business decisions seek only to find an acceptable solution, not one that is necessarily optimal. Despite the difference in nature, the two techniques do complement each other to provide a stronger and more robust solution than either could achieve alone (Armstrong and Hax 1974). A special set of heuristic approaches have been developed in the past (e.g., Armstrong and Hax, 1974; Schriber and Stecke, 1987; Yan and Wang, 2007; Klemmt et al. 2007, 2009; Caricato et al., 2008) that involve both of the methods. One especially useful method is simulation optimization. Research in this area provides an interesting view of the differences that the technologies provide and in what situations it is desirable to combine their capabilities.

2.2 Introduction to Production Ramp-Up

As early as products were manufactured to be sold, producers looked to discover ways to manufacture more goods in a shorter period of time. The faster that they could meet consumer demand, the higher the return they would obtain. Production ramp-ups generally occur whenever a new product is introduced or a company changes its manufacturing operations for a specific product. It can also occur if a company opens a new plant to expand its production capabilities. Production ramp-up is important because it is a risk-filled science that can drastically change the future of a company. Oftentimes, large investments are made in plant and equipment when the company is not positive of whether or not the project will succeed. Unexpected challenges can occur even when adding similar or identical machinery to a plant (Couretas et al., 2001). Different companies may be under different levels of pressure to ramp-up their production depending on their industry, and they may even have different standard operating procedures to ramp-up production depending on the scale of the project.

There is relatively limited research done in the production ramp-up area due to the nature of the concept. The ramp-up strategy for every situation is different, as companies may manufacture the same product using two completely different methods. One ramp-up strategy is not universal, and it may even take multiple methods within different facilities of the same company. Having a strong ramp-up strategy and decision support technology to help with the decisions provides a competitive advantage to the company that is performing a ramp-up of its output. This knowledge and technology can be valuable, and many companies work hard to keep their strategies secret so as to not give their competition an advantage.

Terwiesch and Bohn (2001) describe ramp-up as “the period between the end of product development and full capacity production” (pg. 1). This period is characterized by a high demand for the product but a low production rate and low yield of the new technology due to poor understanding and lack of experience of the manufacturing process. The production rates gradually improve as learning takes place and adjustments are made to both the product and the manufacturing process. Both learning and experimentation in the product’s “recipe” and the process are vital to significantly improving the yield and production rate (pg. 9). The problem of ramp-up often occurs in the high-tech telecommunications industries where products are becoming increasingly complex and are quickly replaced by better, more complex technologies (Terwiesch and Bohn, 2001). These industries are often competitive, so decreasing time-to-volume production is critical to a company’s success and survival.

Many researchers have also found that a lack of manufacturability contributed to poor ramp-up performance (e.g., Koren et al., 1999; Du et al., 2008). In their definition of production ramp-up, Koren et al. (1999) further elaborate to include the importance of quality in a successful ramp-up. They define system ramp-up as “the period of time it takes a newly introduced or reconfigured manufacturing system to reach sustainable, long-term levels of production in terms of throughput and part quality, considering the impact of equipment and labor on productivity.” It is often necessary to make costly changes in the product’s design after manufacturing has begun in order to improve the final product’s quality. A quick response in this case is necessary, but too many engineering changes in design and processing can devastate a project’s success. A high “first time right” ratio will lead to a leaner manufacturing environment with little waste, a high yield and a faster production (Du et al., 2008, pg. 180-1).

All of these ideas that define a “successful” production ramp-up strategy are relative and important to each company that is expanding its manufacturing capabilities. Companies must be flexible and aware of the undertaking because it can become a complicated problem very quickly. Management must know the goals of its individual ramp-up and must tailor its strategy and rate accordingly. There are many techniques that help managers make these important decisions; however, these techniques will not act ideally in all situations.

2.3 Decision Support Systems (DSS)

An important underlying purpose behind studying areas such as production ramp-up is to design techniques and methodologies that can be integrated into accurate and easy-to-use decision support systems for operations management. Understanding how these decision support tools work and how they can be implemented is crucial to building a system that is functional. In order to make strong, well-informed decisions, managers need tools that will quickly provide up-to-date information. Much of the research behind production ramp-up has been implemented into countless decision support tools (e.g., Keen 1980a, 1980b; de Boer et al., 1997; Greenwood et al., 2005; Power and Sharda, 2007; Matta et al., 2007). These tools can be as simple as marginal analysis or a quick mathematical program developed to give a rough solution to a complex problem.

Matta et al. (2007) discuss the importance of using decision support systems in a production ramp-up environment. They found that managing production ramp-up events are related to three issues that justify the need for tools and methods to assist in decision making. First is the identification of the duration of the ramp-up period and all of its associated costs. Second is determining useful tools and methods for managers to make decisions on how to shorten the ramp-up phase. Lastly, they stress the importance that these management tools can help to assess the configuration and reconfiguration of the system. Matta et al. ultimately presented a solution for a capacity-related reconfiguration problem by using Markov decision theory to drive their decision support system. This section provides some background information and history on such decision support systems and how they have been used in industry.

2.3.1 Background

The idea of a decision support system (DSS) began in the 1970s, and they have only become more important to companies as technology has advanced. Peter Keen (1980b) was one of the first to bring to light the importance of the DSS. He published several papers in the early 1980s that demonstrated the advantages of decision support and provided examples of its use. He stressed the importance of recognizing the difference between efficiency and effectiveness when building a DSS. Efficiency is knowing how to do a job, while effectiveness is knowing what job to do. The two concepts go hand-in-hand, and both facets must be satisfied to make a successful decision support tool.

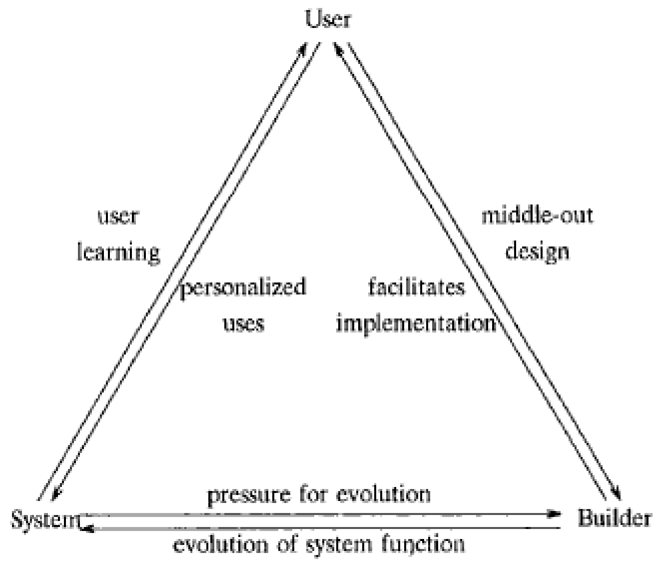


Figure 2-1: An Adaptive Framework for DSS (adopted from Keen, 1980a)

Building a DSS is a dynamic process that requires constant adaptation by the system, its user and its builder. The builder receives feedback from the user and must update the system to adjust to changing or additional needs. The user also receives feedback from the system and the builder regarding the feasible solutions and system capabilities (Keen, 1980a). These relationships can be seen in Figure 2-1. The ultimate goal of the implementation of a DSS is to increase productivity by improving communication, flexibility, learning and responsiveness. The resulting benefits of using a DSS are as follows:

1. “Increasing the number of alternatives examined”
2. “Better understanding of the business”
3. “Fast response to unexpected situations”
4. “Ability to carry out ad hoc analysis”
5. “New insights and learning”
6. “Improved communication”
7. “Control”
8. “Cost savings”
9. “Better decisions”
10. “More effective team work”

11. “Time savings”
12. “Making better use of data resources”
    (adopted from Keen, 1980b)

There are many different types of decision support systems. Many have analyzed and classified different types of systems, determining in what contexts they prove to be most useful. One champion of decision support technology, Daniel J. Power (2007, 2011), provides DSS resources and consulting on his constantly updated webpage <www.dssresources.com>. A summary and description of the different decision support systems are provided below in Table 2-1. These classifications help decision makers determine which type of system could most benefit from their application. Power’s work in model-driven DSS proves to be particularly useful in this thesis.

Table 2-1: Decision Support System Classifications

Decision Support Classification	**Details**	**Example**
Communication-Driven	Collaborative system which has been developed to help a decision maker (DM) identify and solve problems and, ultimately, make decisions using support from another DM or a group	Bulletin board, email thread
Data-Driven	Involves the use of database information from internal and external sources	Data warehouse, ERP
Document-Driven	Revolves around the organization and retrieval of documents in a central location	Lexis-Nexis, InfoSys, UNCOVER
Knowledge-Driven	Computer-based system which relies on data mining to recommend solutions	Database
Model-Driven	Decision support systems which revolve around a model which could be statistical, financial, simulation, etc. and allows for updates and trial-and-error analysis	Decision tree, utility model, Bayesian model
Spreadsheet-Based	Spreadsheet allows for easy manipulation and visualization of data. DSS can be made up of model-driven, data-driven or both.	Solver optimization
Web-Based	Accessible online through a web browser and often incorporates other DSS classifications	Online data records

2.3.2 Applications

With modern technology, the development of powerful model-driven decision support systems in particular has grown rapidly. These decision support systems assist with production ramp-up and other related manufacturing operations. Power and Sharda (2007) analyze the most current applications and research areas for model-driven DSS. They demonstrate the need for more wide-spread use of decision support technology and show how it can reduce costs and increase flexibility of decision makers. They also stress the importance of more state-of-the-art DSS development software. Such software that would allow for multiple models to be developed and run concurrently would greatly increase the complexity of decisions that could be optimized. This type of technology is precisely what is being examined in this research, as few DSS packages allow for mathematical programming and simulation models to be integrated and work in tandem. Power and Sharda challenge researchers in the field to develop more complex model-driven decision support systems that would help increase knowledge about the tools and further research in the field.

Researchers have used DSS technology to support operations throughout all fields. One highly researched area in particular is DSS for naval applications. De Boer et al. (1997) built a system to assist with ship maintenance capacity planning, and Greenwood et al. (2005) used simulation optimization techniques integrated into a DSS package to analyze ship panel manufacturing operations. This decision support system is of particular interest because of its employment of optimization heuristics and a simulation model to produce alternative solutions. The DSS helped decision makers at Northrop Grumman Ship Systems better sequence panel production and improve resource allocation during the process. The user of their system inputs data, such as the production schedule, panel characteristics and job standards. This data set is converted into a form that the simulation model can read. The performance of the system is evaluated, and then the optimizer generates an alternative solution. This process continues until significant improvements are no longer seen. Both the simulation and optimization models run completely in the background with no direct user interaction. The general schematic of the DSS and how it interacts with the decision makers and the actual system can be seen in Figure 2-2. The resulting product of the project was a very simple to use tool, which decision makers could use with little or no knowledge of advanced simulation or optimization techniques. A similarly structured process and integration of the decision support system is used in this thesis in order to solve the production ramp-up problem.

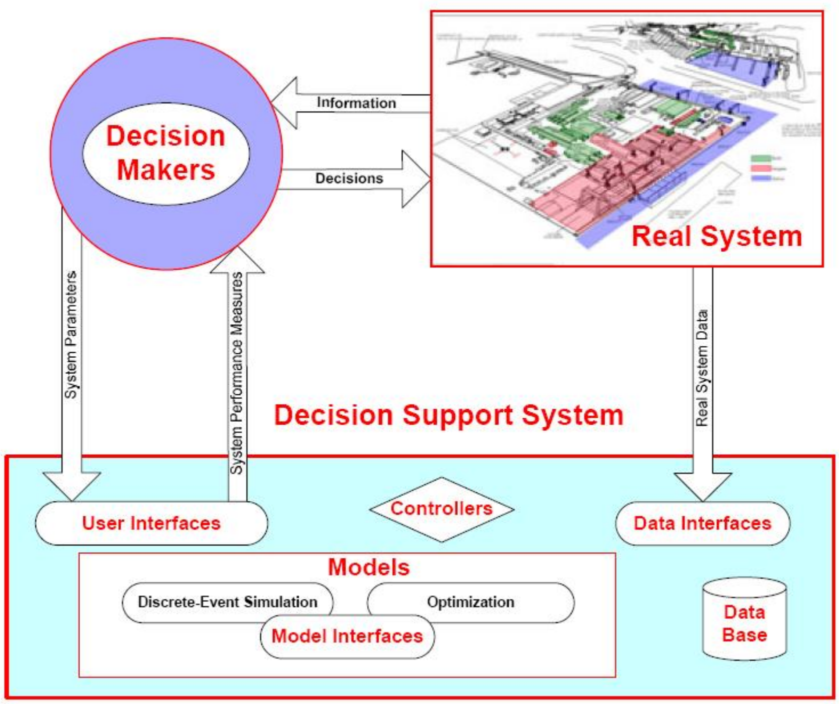


Figure 2-2: High-Level DSS Process Flow (adopted from Greenwood et al., 2005)

Another application of decision support system research that has become more common is that of manufacturing resource planning. Complex decision making tools can help managers to plan resources to meet schedule needs for both the short and long term. Cox and Adams (1980) produced a DSS tool which uses forecasted demand, production plans, the master schedule, material requirement plans and capacity requirements to determine if due dates are attainable. It also produces financial statements for the projected amount of production and calculates the workloads for the equipment. Such a system can require many man-hours to design and implement, but the cost savings can be substantial.

There has been an evolution in DSS research, and the field has grown as manufacturing has become more complex and companies are faced with producing a larger number of more diverse products. Much of the last century's research in the field involved optimization modeling using mathematical programming and a three-step approach. Optimization models first are formulated, an optimal solution is found, and then examination and what-if analysis are performed on the solution. The future of DSS research will find new applications in innovative areas outside of manufacturing, newly surfaced software tools and a newly found appreciation for the tool (Shim et al., 2002).

2.4 Optimizing Production Ramp-Up

2.4.1 Optimization Techniques for One Time Period

Many analytical and heuristic approaches have been applied to optimize ramp-up problems. Some of the most commonly practiced forms of optimization include mathematical programming and queueing theory. These techniques can be especially useful when approaching problems of capacity analysis. This section will discuss some of these methods and how they can be applied to real-life scenarios. There is a wide array of applications for the optimization techniques, and they can be of use in both single and multiple time period analyses.

There is an extensive amount of research performed using mathematical modeling focused on allocating manpower and machines for one defined time period. Dallery and Frein

(1988) use analytical modeling in order to determine the optimal configuration for a flexible manufacturing system (FMS). They describe why they chose the method they did by saying: “The advantage of using analytical models as opposed to simulation is that they are easy to both develop and run. This is attractive for performance evaluation and even more so for optimization purposes” (pg. 208). Their formulation of the closed queueing network minimizes the cost of configuring machines to meet a prescribed level of throughput. They then marginally increased the number of machines in the network for each class of customer and stopped when the desired throughput was reached. The use of mathematical modeling in this case allows for quick marginal analysis. Small models can be run relatively quickly; therefore, many different runs using new inputs can be made and examined to determine what would happen differently with different configurations or numbers of machines. This is one important advantage to using mathematical modeling.

Dallery and Stecke (1990) continued research in the area of server allocation. Their paper determines the optimal allocation of servers into multi-server workstations by using a decomposition method which isolates independent components of a closed queueing network. The theorems that are proposed generate three rules when assigning a flexible manufacturing system’s configuration. The three rules are as follows:

1. For any machine type for which there are no physical constraints, group all machines into a single group (Proposition 1).
2. For any machine type for which the group sizes are prescribed and equal for the different groups, the optimal workload allocation is balanced within this machine type (Proposition 3).
3. For any machine type for which the maximum group size is limited and such that the total number of machines of this type is a multiple of this maximum size, the server

allocation is balanced and each group size is equal to the maximum group size. The workload allocation is balanced (Conjecture 2, pg. 701).

This rule set is useful when determining the optimal allocation of servers in a single class network with a given workload for the system. Its application, however, is limited to this individual instance and the system presented by Dallery and Stecke (1990). The problem becomes more complex when different workloads are applied to the system over different time periods and planning for future configuration adjustments. Such a configuration method is necessary when trying to optimize the configuration as production experiences ramp-up.

Others have also dedicated much of their research to solving machine allocation problems. Boxma et al. (1990) and Frenk et al. (1994) used marginal analysis to solve these machine allocation problems. Marginal analysis, or the evaluation of a particular activity's additional benefits against its additional costs, is often used in correlation with optimization techniques to perform sensitivity analysis. The effects found can often be as important as the solution and can help decision makers better understand the underlying components in their system. Boxma et al. looked to solve the server allocation problem by modeling the manufacturing system as a network of queues. They used marginal analysis to optimally allocate the servers with two goals in mind. First, they kept the work-in-process at a target level and minimized cost by changing the number of servers in the system. Second, they tried to minimize the amount of work-in-process by keeping the number of servers fixed. The algorithms that use this marginal analysis approach were first developed by Fox (1966) and have been used extensively by other researchers including Dyer and Proll (1977), Rolfe (1971) and Weber (1980). Fox's approach begins with an infeasible allocation of machines and adds servers strategically to where they make the largest improvement. Once a feasible solution is found, the algorithm ends, leaving the user with a sufficient solution. This approach does not take into account all of the details of the system, including restrictions on interchanging machines, so it may not provide a completely reasonable solution (Boxma et al., 1990).

Continuing on this line of research were Frenk et al. (1994) who improved the machine allocation algorithm, originally created by Fox (1966), and applied it to semiconductor manufacturing. Similarly, the problem was to determine the minimum cost allocation of machines by still meeting the system's production target. They also provide a worst-case analysis to show the robustness of their improved algorithm. Their algorithm is applied to a semiconductor manufacturing system which was modeled as a queueing network by Van Vliet and Kan (1991) to test their allocation algorithm. Frenk et al. show that their algorithm results in a substantially smaller average relative error than the algorithm proposed by Boxma et al., making it a better methodology to use and apply to other resource allocation problems.

The resource allocation problems can also be examined on a larger scale. Cohen and Moon (1990) and Billington and Davis (1992) researched areas of resource and capacity analysis of entire supply chains in the early 1990s when interest grew after the advancement of flexible manufacturing systems. Billington and Davis carried on Cohen and Moon's investigation into the effects of economies of scale and the optimal logistics and transportation cost on supply chain optimization. They added new decision variables to the linear cost model which functionally open and close distribution centers and determine if a distribution center would service a particular customer zone. Additional constraints were also added to incorporate these decision variables, and this led to a more complete cost analysis of the supply chain. This new model was applied to a case study involving a Hewlett-Packard manufacturing reconfiguration. The model proved its effectiveness when it produced a savings of more than $15 million per year and a solution that senior management had not initially considered. These mathematical programming strategies can help find improved or optimal solutions not only for individual manufacturing systems but also for entire supply chains.

2.4.2 Optimization Techniques for Multiple Time Periods

Optimizing resource allocation during multiple time periods becomes more difficult as more time periods, resources and product types are added. Not only is throughput or work-in-process a driving constraint but so are other periods and their required configurations. Most research in this area is more recent as it often requires faster and more powerful computers to solve problems with large numbers of variables. With an older technology, the use of the mathematical programming techniques becomes less attractive to decision support system creators, since accurate decisions are often needed in a timely fashion. Still there is much less research that has been performed that analyzes this problem over multiple time periods. Very few researchers have examined this problem focusing on resource allocation. Much of the focus has been directed to analyzing the learning curves and the amount of process change involved with production ramp-up (e.g., Carrillo and Gaimon, 2000; Terwiesch and Bohn, 2001; Terwiesch and Xu, 2004).

2.4.2.1 Learning and Process Change

The process of production ramp-up, like any other manufacturing operation under development is dynamic. Production and the evolution of the process are not fixed as the company is trying to determine not only the best way to manufacture a good but also the best way to manufacture more of that good. During this period of scaled-up production, many deviations from the prescribed method of production occur. Determining quick solutions to problems is important in order to maintain a certain level of production, and these solutions are not always planned for but are necessary. These discrepancies can change the production process from how it was intended and first planned to operate but are more difficult to avoid with larger ramp-ups.

Terwiesch and Xu (2004) define the reduction of these discrepancies as the process of learning. This process is a necessary part of any ramp-up, and its goal is to improve the yield and output of the production process. There is also another component to learning, process change, which is often necessary but counterproductive. Companies look to improve their processes by changing them. The end goal of process change may also be to improve yield and output, but more learning is necessary to understand the changes and reduce discrepancies from the way the process was originally designed.

Terwiesch and Xu (2004) developed a non-linear model which had been studied previously by researchers who explored the area of learning, and they applied it to production ramp-up. The deterministic, profit-maximization model uses the company's production rate, amount of process change and learning effort over time to maximize profit over the production ramp-up period. The model is closely related to one generated by Carrillo and Gaimon (2000) that also takes into effect the detrimental effects of process change. Their non-linear model as well as that by Terwiesch and Bohn (2001) uses capacity loss of the company to model the effects of process change. While the company loses capacity for the short term, it gains knowledge. Terwiesch and Xu's new model does not assume the company gains knowledge just by changing the process, but it allows the company to reach a new frontier after additional learning takes place. Due to the detrimental effects of process change to a production ramp-up period, Terwiesch and Xu propose a "copy-exactly" strategy and propose some characteristics under which it will work best. Under this type of strategy, a company implements the exact production process that it used during product development in the full scale manufacturing of the product.

The results of the Terwiesch and Xu (2004) model determined several important characteristics that are important to companies who plan to ramp-up using a strategy of least process change. First, ramp-up processes with little starting knowledge would benefit from a copy exactly, or "CE-ramp," because of their ability to quickly learn with small investments. Ramp-ups that are difficult to improve or very sensitive to change would also benefit because of the high cost incurred when changing any aspect of the system. Lastly, ramp-ups with shorter lifecycles or steeper demand growth will also benefit from a CE-ramp. These types of products benefit because it is more important that yields are kept as high as possible during the short time period when demand is high. At the same time, process change is delayed so as to not disrupt the shorter time frame, which holds a higher opportunity cost. Although it may seem impossible to avoid process change, in some ramp-up scenarios it is important and beneficial to avoid or delay process change as long as possible. Oftentimes, this technique is employed in the high-tech, semiconductor industry where product lifecycles are short, demand growth is steep and process change is expensive.

2.4.2.2 Resource Allocation

Other researchers have focused less on learning and process change during ramp-up and more on equipment and machine expenditure. Huang et al. (2006) wrote: "Equipment expenditure generally accounts for a substantial portion of total investment capital in a typical job shop. Therefore, manufacturing companies… must deal with the important problem of how to allocate and expand the processing machines used in their job shops in a step-by-step fashion" (pg. 148). It is often too expensive to upgrade all equipment at once, and job shop management must decide which piece of equipment to invest in first. They must bring the equipment online in such a way that will also not disrupt the flow of production. Allocating machinery incorrectly can create bottlenecks, idle machinery, early depreciation, a large amount of work-in-process and poor efficiency, all which create poor manufacturing and financial reports.

The Huang et al. (2006) approach seeks to meet specific performance targets for each period of the ramp-up process while minimizing machine investment. The integer-planning formulation that they developed explores each period individually and uses the previous period's workstation allocation strategy as a starting point for the new period. Huang et al. term this strategy the "forward recursive scheme with improved marginal analysis." It essentially optimizes each period individually running an adjusted math program for each new time period of increased production. This technique helps produce a solution very close to the optimum. Their strategy is applied to a ramp-up problem at a new virtual semiconductor manufacturer with 78 workstations and produced savings of over $17 million in machine procurement costs. The conclusion drawn from their study was that the solution technique was very practical and could be applied to manufacturing systems that are even less complicated than the semiconductor manufacturing problem.

When analyzing a production ramp-up problem, it is critical to take into account multiple time periods. By the nature of the problem and how manufacturing companies ramp-up production of their products, a step-by-step progression of machine expansion is necessary. One large investment in the beginning would be unreasonable and would produce a large amount of excess machine capacity before the company could catch up with other aspects of the production system. Mathematical programming techniques, including linear and integer programming, can be used to optimize each period individually, as was the strategy used by Huang et al. (2006), or it can optimize all periods simultaneously, which can help with the synchronization between time periods. Math programming techniques are not the only ones that can help optimize the machine expansion scheduling problem, but they are also one of the most widely used.

2.5 Simulating Production Ramp-Up

Despite their similar uses and shared goals, simulation is not optimization. Simulation has been widely used for many different manufacturing applications especially as computer power improves and the number of simulation software packages increases (e.g., Harmonosky et al., 1999; Couretas et al., 2001; Rosen et al., 2008; Eklin et al., 2009). It can be used to determine or predict the performance of a system. It will not tell the simulation analyst how to adjust the system to make it better. Some software packages will incorporate optimization tools, but simulation is generally not regarded as an optimization tool (Armstrong and Hax, 1974). Its usefulness often comes in how it can be easily manipulated to perform marginal analysis on a manufacturing system. The simulation will update statistics that can help an analyst determine beneficial changes to the system. It will also show how the system will perform under stochastic conditions based on user-selected distributions for different variables.

Production ramp-up is a dynamic process where the manufacturing system is constantly changing to increase production to a higher level. Significant stages during the ramp-up process, including the initial and final states, can be modeled to simulate production during different phases of the ramp-up. This can be helpful during long-term ramp-ups or those with progressive stages throughout. Manne (1961, 1967), who was one of the pioneer researchers in the area of capacity planning and expansion, took this very approach of planning a ramp-up based on regular intervals. His first paper in 1961 discusses capacity planning for a plant and how to optimally equip the facility with excess capacity for future use. Manne does this by producing a probabilistic model that minimizes an expected cost function. In his subsequent paper, he determined that the optimal ramp-up policy for linearly increasing demand is for a plant to add machinery at specific intervals and that this will lead to the optimal plant capacity for a given demand. Much of the focus of his papers was in the distribution of the demand, and he demonstrated the importance of the demand as an input to the problem. Researchers after him have tested the problem using different demand growth distributions, showing the robustness of his optimal plant growth policy (e.g., Couretas et al., 2001; Klemmt et al., 2009).

2.5.1 Simulation as a Performance Evaluator

Determining the performance of a production ramp-up and its associated fixed asset costs has been generally approached using static techniques. Many production managers find it difficult to learn a simulation language and resort back to these inflexible methods or simple queueing methods of solving capacity-related issues. However, the evaluation of a manufacturing system's performance and how it is correlated with demand on the system requires a dynamic evaluation technique. This is where simulation has proven useful. When system inputs such as resource availability are variable, techniques such as linear programming and queueing theory do not prove to be as useful or as accurate as simulation. Using simulation for such performance evaluation methods also allows the demand and other input factors to be tested both on discrete and continuous scales (Couretas et al., 2001).

Manufacturing cost is an important performance metric for most companies. Since a cost can be assigned to different parts of an operation, many companies use cost as a metric to determine if a project or configuration is feasible. Beginning in the 1990s, simulation helped many companies make more accurate estimations of operations costs. The use of simulation allowed for better automation of the process and increased the popularity of activity-based costing. Using simulation, processes could now be simulated activity-by-activity, making movements and actions more visible (Takakuwa, 1997). Costs can then be assigned to each of these motions, which allowed for more accurate estimations of cost.

As the importance of activity-based costing was seen across many industries, economic value added (EVA) analysis became another important tool to measure profitability. EVA is especially important to a production ramp-up problem because it takes into account the total cost of capital, and capital expenditure is generally the largest cost incurred during ramp-up. EVA also helps tie in the different business divisions' roles and is a good indicator of how management is performing. It is regarded as especially important to shareholders as it is an indicator of cost of equity capital, and it attempts to match the timing of capital expenditures with the resulting payoffs (Couretas et al., 2001). This component is especially important as shareholders look to see how long the payback horizon is for any large expenditure. The longer the horizon, the higher the risk involved. Shareholders are less willing to fund projects where they will not see their money back until 10 or more years down the road. This demonstrates the importance and need for a quick and efficient ramp-up procedure that will quickly produce return on investment.

It is particularly important to use key performance evaluators when using a dynamic simulation to analyze a manufacturing system. The performance measures can easily be obtained as statistics of the model and will help validate the model's functionality. The Couretas et al. (2001) model was developed from extensive research from Zeigler (1984, 1990), Zeigler and Praehofer (2000) and Couretas et al.'s prior work in 1998 and 1999. It was built to exhibit simulation-based manufacturing capacity analysis. This analysis found an optimal balance between manufacturing capacity and work-in-process. It used performance measures such as return on net assets (RONA) and return on operating assets (ROOA), which take into account short and long-term capital investment strategies. Measures such as these help evaluate solutions in the simulation model and create local optimal solutions. The model created by Couretas et al. (2001) used quarterly demand level, fixed machine costs, failure/repair times, processing times, processing yield and administrative polling time as input parameters for the model. These parameters fed the simulation model which presented the user with outputs such as machine availability, line utilization, cycle time, work-in-process, RONA and ROOA. Two different approaches were considered: "optimistic" and "conservative." The optimistic approach liberally adds capacity, whereas the conservative adds just enough to produce the period's projected demand.

The conclusion from Couretas et al. (2001) is that simulation provides a useful but limited instrument for solving capacity planning problems. The method was tested using only one product stream and assumed that instantaneous additions of work cells could be made. It was also a considerably complex model that the authors found to be difficult or impossible to solve without a program supporting distributed simulation. The results show that the method supports Manne's suggestion of adding capacity at constant, defined periods. This "conservative" technique provides the adequate capacity expansion with lower overall investment for the company (Couretas et al., 2001), and the methodology provides a general underlying structure that is applicable to many different optimization problems. Their research brings up many new directions and demonstrates the functionality of simulation in quantifying production ramp-up performance.

Like the model developed by Couretas et al. (2001) that judges a system's performance based on RONA, ROOA, utilization, cycle time, etc., Rosen et al. (2008) demonstrate the importance of analyzing system designs using multiple performance measures. Using only one performance measure to assess system performance, an area that has been researched extensively, can limit a method's ability especially when more complex models with conflicting performance measures are involved (Rosen et al., 2008). Such models cannot be formulated using a simple cost function. Rosen et al. provide many suggestions about new approaches to solve multiple objective simulation problems. One suggestion was to include simulation optimization techniques, which better incorporate the user's risk and uncertainty preferences. This problem could be approached by using different methods including the multiple attribute utility method, which uses a utility function to capture the end user's preferences, or a metamodeling approach such as response surface models or neural networks, which would provide a non-dominated solution and allow the user to evaluate the solution based on personal preferences. Rosen et al.

show that this area of simulation is still wide open with new topics to be developed allowing for more complex simulation tools and applications.

2.5.2 Simulation-Based Marginal Analysis

Simulation can also prove to be a useful tool for marginal analysis of a system's performance. The addition, subtraction and rearrangement of machines, the workforce and the overall work flow can easily be manipulated before a large capital investment is made. This tool has widely been used by production managers to test the performance of their systems and see where they can make the largest improvement with the smallest change. Many different scenarios and combinations can be tested in a small amount of time allowing for greater probability of finding a sufficient solution. Most simulation languages can also accommodate small details in a system, which makes the model highly accurate. Marginal analysis is an important procedure in examining the additional benefits of a change to the system compared to the additional costs. The use of simulation has only broadened the capabilities and ease of performing this analysis technique.

Though not much research focuses solely on marginal analysis, it is generally used to compliment optimization problems or the analysis of production systems. Rubinstein (1989) provides a strong background into the methods for using simulation and sensitivity analysis of a particular performance measure. All of the sensitivities and performance estimation methods that he produced can be made straight from the simulation model itself. Harmonosky et al. (1999) showed how marginal analysis helped save money during the ramp-up of high density interconnect (HDI) modules for Lockheed Martin Government Electronic Systems. Their simulation model interfaces with costing software to help make the best decisions when designing the higher-capacity lines. Eklin et al. (2009) also utilize marginal analysis when developing their model that also optimizes a shop floor layout using costing estimation. Different researchers have used similar techniques of optimizing diverse system using marginal analysis and applying these methods to unique case studies that show its extensive applicability.

The method that Rubinstein (1989) presents, the "score function (SF)" method, was unique in that it simultaneously estimates the driving objective value as well as its associated sensitivity information. He justifies the method's practicality by saying that most people who are trying to optimize a given system are interested not only in its optimality but also its sensitivity to changes. The sensitivity components that he studies in particular are the gradient and Hessian metrics. This makes it a realistic methodology to follow for systems, such as reliability systems, stochastic networks and queueing networks. He also demonstrates that the performance measure for his model, the efficient score, is simpler and requires less computation than a general performance estimate. The Rubinstein model is also open to improvements including several variation reduction techniques. This research drew on much of the earlier research on performance measurement and sensitivity analysis (e.g., Law and Kelton, 2000) and helped develop new applications for simulation and marginal analysis.

Harmonosky et al. (1999) focused on the interaction between cost-optimization software and the flexibility of simulation modeling. Cost is often the driving performance measure for many simulation optimization problems, so the interaction between discrete event simulation and costing software, which can estimate costs for auditing and proposal purposes, can be useful and applicable across many industries. The costing software, however, cannot accurately estimate costs for a manufacturing ramp-up problem that incorporates variability, competition for resources and material handling operations. The costing software provided baseline inputs for the simulation model which produced extensive output to calculate total costs using explicit cost equations. The model was then run for different scenarios that altered either the inter-arrival time of parts, the number of workers, the number of skills per worker or additional equipment. The resulting work-in-process, labor, scrap, tardiness penalty and total costs were calculated using the derived cost equations to compare the scenarios and find the optimal configuration. Though the cost and throughput values may change, the report provides a method for sensitivity analysis that is applicable to many different scenarios. It also demonstrates the practicality of using an optimization technique to feed a simulation model. This gave their analysis more flexibility to handle the system's inherent variation and created a more robust solution.

Eklin et al. (2009) took a different approach to solve a similar capacity-related problem of a shop floor. Unlike other research, the model they developed takes into account the stochastic behavior that is associated with a shop floor. The simulation model also uses cost estimation to determine the optimal shop floor layout. Marginal analysis is used to measure the difference in total cost after adding one additional order. The technique used in the report runs a simulation model to test a linear program's output and then recycles the data back to the optimization procedure until a suitable feasible solution is found. Their solution method improves on a decision support system developed by Feldman and Shtub (2006) that helps managers decide whether or not to accept a new product order. Their model minimizes cost as a function of load on the shop floor. It follows a three-phase method that first optimizes the production volume of each product on the floor. It then tests the feasibility of the solution by using a scheduling procedure. If the solution is sufficient, it is accepted. If not, the capacity of the shop floor is adjusted marginally, and the problem begins again from phase 1. Eklin et al. suggest several improvements to the model including estimating the cost of orders marginally rather than by total cost and replacing the scheduling procedure with a simulation model. The process that their heuristic model follows can be seen in Figure 2-3. Their method improves the solution in terms of cost as well as CPU time. Their method was more accurate in its cost estimation because it takes into account not only processing time, but also idle machine time as well. The model also limits the amount of time a product can remain in the system to improve throughput and eliminates a weakness of the original procedure that could lead to missed feasible solutions by making too large of a capacity update between subsequent runs.

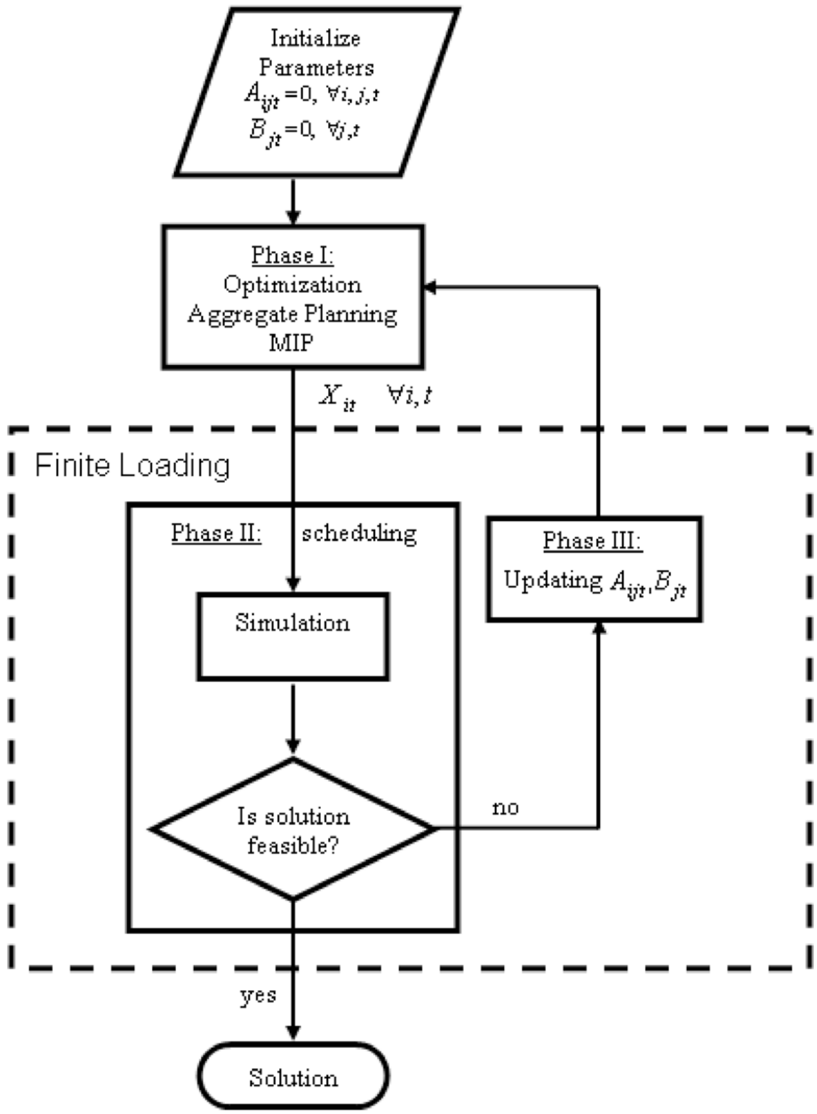


Figure 2-3: Process Flow Diagram of Heuristic Model (adopted from Eklin et al., 2009)

The use of costing software and other programs to optimize a system or an equation provides the first look at techniques of solving simulation optimization problems. One such technique is using mathematical programming to produce the input needed to feed the simulation model. This similar approach was seen in the method used by Harmonosky et al. (1999) to create a robust solution to a capacity expansion problem and by Eklin et al. (2009) to optimize shop floor capacity. In the following section, this topic of utilizing optimization techniques with simulation and the current research methodologies will be discussed in more depth.

2.6 Simulation Optimization

It is important to emphasize the difference between simulation and optimization. Simulation models are built to provide a detailed, realistic solution to a system that may be too complex to write a mathematical programming for or use another optimization technique to solve. The simulation will not provide the optimal solution. Mathematical programming models will, on the other hand, provide the optimal solution, but they are not subject to any variation. This allows optimization techniques to generate "optimal" values that can be used in a simulation model of the system under investigation. The simulation model then can determine how robust the solution is and if the solution is optimal or close to optimal in a more detailed model. Although these two techniques are often regarded as the two most widely used operation research tools, the combination of them on a large scale is not yet available (Fu, 2007).

Simulation optimization is a subcategory of optimization that is becoming more popular as manufacturing systems become more complex and difficult to simulate. A simulation optimization problem can be defined as an optimization problem where the constraints, objective function or both can only be evaluated using computer simulation (Caricato et al., 2008). The problem cannot be solved using standard analytical techniques. Even though the field has been rapidly growing with new software available and faster computers to run more complicated models, it is important to create optimization algorithms that are as efficient as possible so they solve in a reasonable time.

Fu et al. (2008) provide a tutorial of simulation optimization, offering approaches and discussion of several areas of research. Approaches to simulation optimization problems include sample path optimization, sequential response surface methodology, stochastic approximation and deterministic metaheuristics. Sample path optimization, also known as sample average approximation, seeks to use a large enough sample size to eliminate the stochastic nature of the problem. This way, non-linear programming tools can be applied. Sequential response surface methodology provides algorithms that utilize statistical methods, such as regression, to search the feasible space for superior solutions. The stochastic approximation method uses algorithms that mimic gradient methods in non-linear optimization to converge asymptotically on an acceptable solution. Lastly, the deterministic metaheuristic category includes many approaches including genetic algorithms, tabu search, scatter search and other iterative algorithms. These techniques involve little statistical considerations, but prove to be useful in solving simulation optimization problems.

2.6.1 Simulation Optimization Techniques for Scheduling Problems

The scheduling problem is one that all production systems face, and solving it is critical to organized operations and reaching potential production capacity. Many researchers have looked into this problem in the past, solving it using different methods, and it has been explored in more depth than the production ramp-up problem. As far back as 1959, mixed-integer programming (MIP) techniques were used by Wagner (1959), Bowman (1959) and Manne (1960) to solve small scheduling problems. These solvers were not particularly relevant because of their small size, and adding more objects makes the problem grow very rapidly to an unsolvable scale. In recent years, a movement from these mathematical solving techniques, such as branch and bound, to more heuristic approaches, such as simulation-based optimization, has taken place because of its flexibility (Klemmt et al., 2009). New computing power has allowed for more exact scheduling solutions to be produced in a shorter amount of time. Recent research has revealed the benefits of using simulation optimization to solve real scheduling problems that modern technology makes manageable.

Much research was performed by Pinedo (2009), who developed and popularized many mathematical programming techniques to solve job shop scheduling problems. He demonstrated how the problems could be formulated as both mixed integer and disjunctive programs. Pinedo preferred to use disjunctive programs as they could initially take into account some ordering of jobs on a single machine. These mathematical formulations as well as Pinedo's research in genetic algorithms helped Klemmt et al. (2007) develop their simulation optimization technique which proved to be both highly flexible and accurate. Their "global iterative operating (meta-) heuristic" is a cyclic approach that first runs a simulation model and then updates an objective function value using a heuristic optimization algorithm. A diagram of this cyclic approach is shown in Figure 2-4.

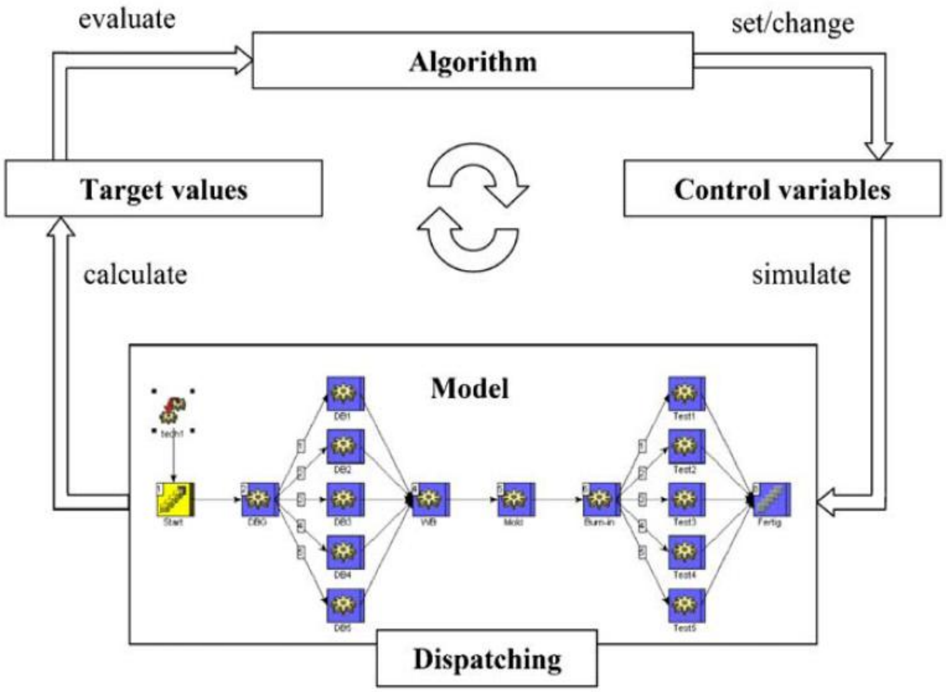


Figure 2-4: Simulation-Based Optimization (adopted from Klemmt et al., 2009)

Klemmt et al. (2009) used a more generic mixed-integer programming (MIP) formulation when applying his simulation optimization approach to a job shop scheduling problem. The advantages and disadvantages of using a solver and a simulation approach are discussed and summarized in the Table 2-2. Using simulation in solving the scheduling problem allows for the simpler MIP to be used, and its interface allows the user and an underlying enterprise resource (ERP) system to quickly and easily enter the large amount of data needed to run the simple math program. The MIP model takes into account additional information that restricts the optimal solution, including release dates, due dates and setup data. The math model, which generally requires more training to develop and understand, is automatically generated for the user from the simulation. This makes the tool a very easy-to-use decision support tool that can solve practical problems because of the reduction in variables created from unique index set definitions. The system developed is flexible for job shop scheduling, especially in real-time, and can also be interfaced with most simulation languages with only slight modifications.

Table 2-2: Comparison of Simulation-Based and Solver-Based Optimization (adopted from Klemmt et al., 2009)

Method	Advantage	Disadvantage
Simulation-based optimization	• Simple modeling (also for complex problems) • Resource constraints modeling • Problem specific extensible via additional scripts • Automated model generation possible (from ERP system)	• Slow convergence • Not necessarily optimal • Implementation of time related constraints
Solver-based optimization	• Fast convergence (for small problems) • Time constraints modeling • Exact solution possible (incl. proof)	• Only small problems exactly solvable • High modeling effort (know how)

A similar approach was used to optimize a job shop scheduling problem by Yan and Wang (2007). The simulation model they developed evaluates the performance of proposed scheduling plans and feeds the results back into a genetic algorithm module, which then formulates the next generation of candidate plans to be tested. The genetic algorithm module that is used guides the system to superior solutions using a series of scheduling rules. This cycling of information between the genetic algorithm and the simulation model continues until a satisfactory scheduling solution is found. Their simulation model incorporates control logic from shop level, cell level and the equipment level, which is often overlooked, but adds flexibility and reusability to their model. They show from their simulation model that it is advantageous to assign individual scheduling rules to each machine rather than an identical scheduling rule to all machines.

There has been a wide array of research performed on optimization modeling of scheduling problems. It is particularly useful to examine such techniques that were applied to a job shop application. It is clear that similar techniques can be applied to optimization of other aspects of a job shop manufacturing system.

2.6.2 Simulation Optimization Techniques for Resource Allocation Problems

Simulation optimization techniques have proven to be useful in solving scheduling problems, but they can also be useful when determining the optimal allocation of resources in a production system. Resource allocation problems look to determine the optimal number of machines, workers, tools, inventory, etc., in order to achieve a certain performance measure such as throughput. The resource allocation problem can be equally important, if not more important than the scheduling problems that a manufacturing floor faces. Without an adequate number of machines to perform all necessary jobs, it is inevitable that products will be left unfinished or machines will have excess capacity. Decision makers are often more interested in resource allocation because the investment associated with it is generally substantial, and it can drastically affect the outcome of a project from the start. There is a limited window of opportunity to purchase the correct quantity of machines at the beginning of a ramp-up project, whereas the schedule can be adjusted periodically depending on performance as well as new product requirements.

Though simulation is often used to analyze systems with high complexity involved with the number of components and their interactions, techniques have been proposed to map a discrete event system into a mathematical program. This alternative method of modeling was proposed by Schruben (2000) who represented the system as an optimization model minimizing the sum of event times, which attempted to execute each activity as early as possible. His model was constrained by the routing of customers, limited buffer capacity and waiting times. This model was extended further and analyzed for single-server tandem queues by Chan and Schruben

(2003). They demonstrate how their model, which they solve using the linear model's dual problem, can be also applied to multi-server tandem queues as well. They also later examined how discrete event systems could be modeled by using the structural properties of systems and how to route entities through the shortest network paths. Matta (2008) researches this with particular interest in how to optimally allocate resources in a production flow line system. She develops three types of mathematical programming representations, including a mixed-integer linear program, a linear approximation model and a stochastic programming model of a discrete event system that perform the optimization while many of the performance measures are calculated using standard queueing equations. It is determined that the linear approximation model proves to be a useful fast search technique when trying to find more optimal areas of the solution space, whereas the stochastic programming model is more exact but also limited to small search areas. She applies her models to a production flow line problem, but it can also be applied elsewhere such as kanban, CONWIP and assembly systems.

The problem of flow shop configuration was also solved using discrete event simulation and mathematical programming tools in a method designed by Caricato et al. (2008). Their method centers around a multi-objective math program that attempts to minimize both the number of workers required to maintain the flow line and machine underutilization. Due to the different nature of the objectives, it is difficult to assimilate the two functions into a single objective, such as a cost function. In order to solve the problem, an integer program was formulated to optimize the second objective using every possibility of the number of workers that could be assigned to the line. The best solutions were set as inputs for the simulation, which provided the non-dominated, Pareto-optimal solution. The hybrid technique, shown in Figure 2-5 below, was tested against the commercially available simulation optimization software, OptQuest, and proved to provide a superior solution. It also solved in less time than the OptQuest set because the proposed method was able to limit the number of simulation runs to those which would provide suitable solutions.

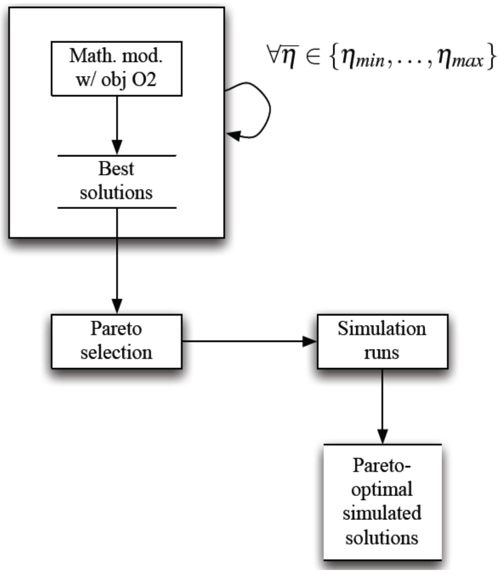


Figure 2-5: Model Overview where η is Number of Workers (adopted from Caricato et al., 2008)

Solving resource allocation problems using simulation optimization is common due to the complexity of different types of modern manufacturing systems. Such a technique has been used to analyze job shop production systems for several different applications first by Armstrong and Hax (1974) and then in a flexible manufacturing environment by Schriber and Stecke (1987). The simulation optimization technique was endorsed early on by Armstrong and Hax to analyze the design of a naval tender job shop. The goal of their model was to optimally configure the job shop's machines and workers by minimizing cost. They also wanted to determine if the job shop should invest in numerically-controlled (NC) machines and how many should be purchased. NC machinery was a new technology at the time of the paper's publication, so it was a large investment for the company seeking to upgrade its manual lathes and machining centers. The model did take into account multiple time periods; however, demand remained relatively steady throughout, so it was not a ramp-up problem. After the optimization of the mathematical model, the simulation model was then used primarily to test how the configuration functions with the different product routings in the job shop. Based on the nature of a job shop, there are many products with different sequences of processes. Simulation proves to be very useful when testing many scenarios with differing levels of uncertainty. It also allowed Armstrong and Hax to ensure that their mathematical model was correct by running the simulation with the manual and NC machine configurations of interest.

Schriber and Stecke (1987) used mathematical programming and simulation to determine the optimal level of resource allocation which would maximize machine utilization in a flexible manufacturing system. Simulation provided a means to test certain factors of the system not incorporated into the optimization program due to the difficulty in accounting for them. These variables tested in the simulation model include secondary resources, physical layout limitations, secondary time requirements, operating procedures, operating interruptions and job characteristics, such as part routings and due dates. Lastly, the sensitivity of the machine utilizations was tested to determine the flexibility of the FMS operations.

The importance of simulation optimization is evidenced by how far research in the area dates back. Even when the power of current computing and modeling systems was minute compared to the technology available today, methods of simulation optimization were being developed. It was clear that mathematical programming or simulation alone could not solve all of the scheduling and resource allocation problems of even moderately complex manufacturing systems. Many methodologies have demonstrated the benefits of using simulation optimization to model and assess all different types of production systems. It eliminates the classic trade-off between accuracy and flexibility and determines the optimal solution that is both exact and robust.

2.7 Summary and Research Goals

This literature review has provided an introduction to the concepts and methods which have been used to solve production ramp-up and capacity-related problems. Researchers have used many different techniques to solve the problem which can be critical to a manufacturer's success. Large investments in capital are to be made, and if the decision support technique is not adequate, the ramp-up process will not produce the optimal output. Techniques such as mathematical programming have long been used to solve such problems, and simulation has proven in more recent years to be a useful tool for complex manufacturing systems. More recently, simulation optimization techniques have been applied for problems in which the system is too complex for one of the techniques alone. Search heuristics, such as genetic algorithms, and marginal analysis have been incorporated into simulation optimization techniques to add to their flexibility and accuracy.

This research aims to develop a simulation method for optimizing resource allocation during production ramp-up in a job shop. Much research has been done in the area of production ramp-up, from learning curves to capacity analysis, but few, if any, have analyzed resource allocation in a job shop. Research in the area is generally regarded as company specific, but generalizations can be made that apply to production ramp-up of job shops of any type or scale. Conclusions drawn from this research and the system that is developed will be applicable to manufacturers looking to increase throughput in job shop operations.

Chapter 3

Methodology

3.1 Problem Statement and Objectives

With the increases in competition between manufacturing companies to constantly deliver new, innovative products to the expanding global marketplace comes the growing importance of efficiently ramping-up production from the first unit to full capacity. Definite ramp-up strategies are not universal and cannot be applied across all manufacturing systems. This thesis will look to determine the optimal ramp-up production strategy in a job shop environment with multiple workstations and product lines. Given product demand by time period, the optimal number of machines required to produce the given throughput will be determined when considering purchasing costs of machines, operating costs and the cost associated with unfinished products. There are also many constraints that can further complicate the problem including financial, spatial, labor and other special case limitations. The ramp-up strategy in this case will be evaluated by several different performance measures including cost, unfinished products, workstation utilization, average time in queue and average time in system. These measures provide a wide array of information to the decision makers and will help determine the best ramp-up strategy.

The main objective of this thesis is to develop a robust solution procedure to determine the optimal ramp-up procedure for a job shop manufacturing system. It will utilize both optimization and simulation techniques that work jointly to provide this solution. The solution procedure will also allow for manipulation and input from the user who can set target performance measures and marginally alter the solution until it is acceptable. Marginal analysis in this system will focus on analyzing the benefits of making a change in the system to the additional costs of that change. The end user makes the final decision as to whether the benefits outweigh the costs and that the change should be realized. This part of the solution procedure is crucial when multiple criteria are to be considered. In contrast, sensitivity analysis will be performed to show the sensitivity of the output to changes in the input. Rather than adjusting the solution that is provided by the system, sensitivity analysis examines the effect of changes to the system's input parameters. This allows the user to test input parameters in the system which may not be precisely known and to determine their effects on the system's output. After the solution procedure and models are explained, a case study will be solved to display the system's capabilities.

3.2 Assumptions

Several assumptions were made about the model being tested in order to simplify the models and make experimentation possible. Those assumptions are listed below and describe the job shop discussed in the following chapters:

1. Machines can only be added at fixed intervals between time periods. They may not be introduced to the system during the duration of any time period.
2. All time periods are of equal duration.
3. All machines in a given workstation are identical throughout all time periods. Machines purchased in a later time period do not experience any technological changes that would increase their efficiency.
4. The number of machines or products cannot be fractional. All machines and products are considered to be integer.

5. Machines cannot be sold or removed from the system during the ramp-up phase.
6. If a machine is utilized at any point of a time period, its associated operating cost for the period is applied to the total cost of the ramp-up phase.
7. Machines do not experience failures or breakdowns. Machine failures and repair time are not accounted for in this study.
8. Demand is deterministic with no uncertainty about its quantity or timing. It is constant over the entire time period and is introduced into the job shop at a constant rate.
9. The product mix remains constant throughout all time periods.
10. No unfinished products are in the system when the ramp-up period begins.
11. Products that are unfinished in a given time period are given priority to be finished during the following time period. Their quantity is added to the new demand for the subsequent time period.
12. It is assumed that there are no spatial, labor or material handling limitations on the job shop system.

3.3 Solution Procedure

An algorithm was created to test the production ramp-up system and arrive at an optimal solution that the user can adjust marginally. This algorithm describes the process that the system goes through in order to reach this solution and sets the criteria which the system must meet for the job shop being simulated. These criteria are essentially acceptable limits for the critical performance measures of the system. The individual user of the system can choose specific criteria that apply more consistently to his or her individual situation. A process flow chart of the algorithm's solution procedure is shown below in Figure 3-1. An example of the process is then demonstrated in the following chapter where criteria were generated to resemble that of a typical job shop system.

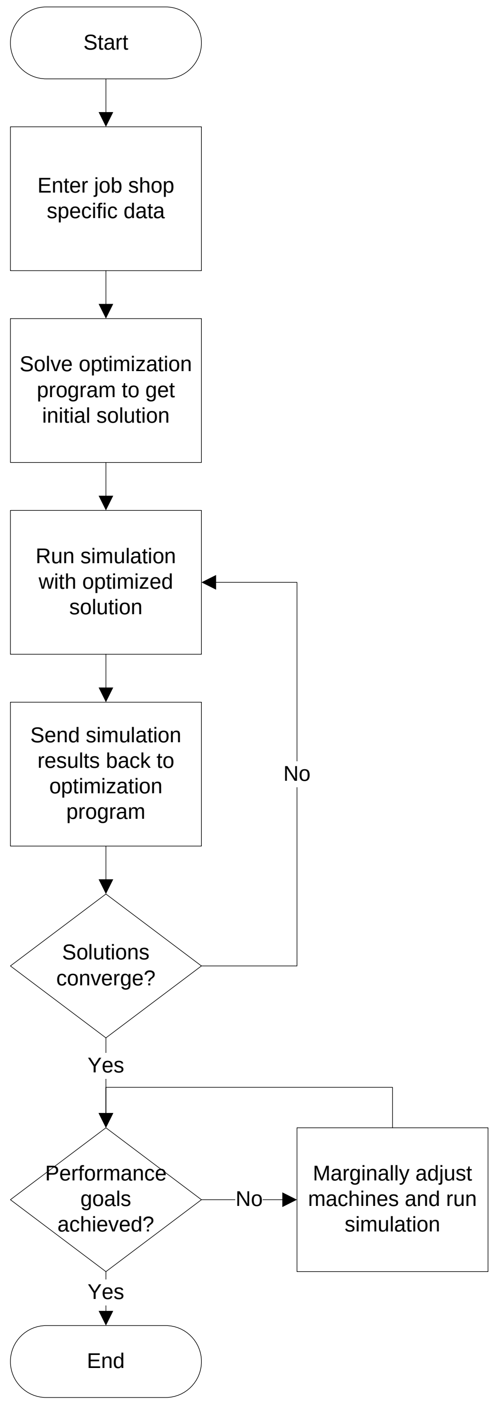


Figure 3-1: Flow Chart of Algorithm Procedure

The first step in the algorithm is for decision makers to enter system-specific data into the system in order to tailor it to their own job shop environment. More detailed and accurate data entered into the system will yield a better output from the system. These data are then used in the optimization model to quickly generate an initial solution that is close to an acceptable solution. The result is run through the simulation model to test it under stochastic conditions. This tests how the system performs with the optimization program's solution, outputting key performance measures including how many unfinished products were left during each time period. These data are then fixed in the optimization program, which is run again. The process repeats until the solutions converge and an optimal or near-optimal machine ramp-up configuration is achieved. If the decision makers have specific performance goals, like cost or a desired utilization, they can marginally adjust the solution until their desired performance goals are met. Doing so, however, may make the solution less than optimal. A more detailed view of the ramp-up system's processes and interactions can be found below in Figure 3-2.

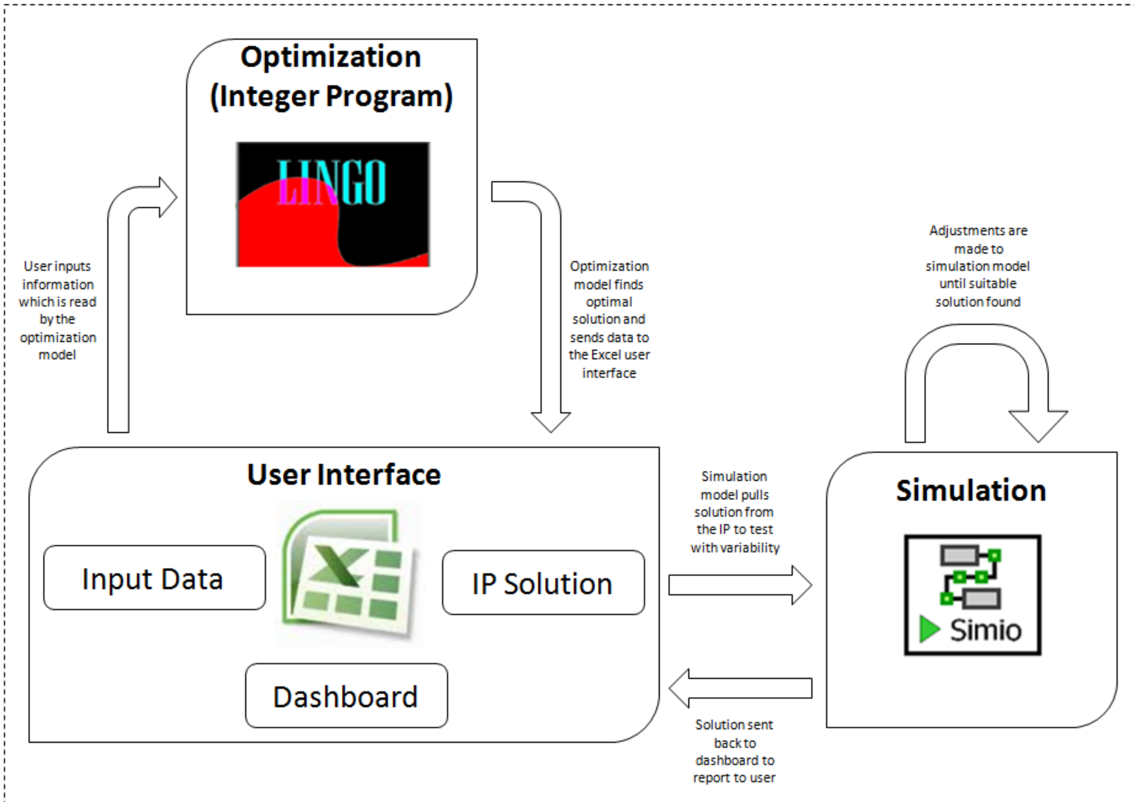


Figure 3-2: Detailed View of System Functionality

The production ramp-up system is made up of three main components. These include the user interface, optimization model and simulation model. Initially, decision makers enter the system parameters that they would like to test. This information is then sent to the integer program to quickly determine a near-optimal solution. This solution, however, has yet to be tested in a stochastic environment. The idea behind this is to efficiently produce a solution in a time that would be much shorter than it takes a typical simulation optimization software. This solution is reported and displayed to the decision makers in the user interface and then passed on to the next leg of the system to test its robustness.

In order to test the solution's robustness and how it will align with the acceptable criteria, the solution is run through a simulation model. The integer program's solution is passed to the simulation model, which will determine how the system will actually perform in the presence of variability. The simulation model and the optimization model pass the solution back and forth until the solutions converge, and an optimal, robust solution is found. The resulting performance measures from this solution are compared to the acceptable criteria set by the user. The model can then be marginally adjusted by adding or removing machines to workstations that do not meet the user's specifications, and the simulation run is repeated. The suitable solution is found when all of the solution criteria have been met.

This has been a general overview of how the ramp-up system works and some of its benefits. In the following sections, the integer program and simulation model will be described in greater detail.

3.4 Mathematical Model

The purpose of the mathematical model in the job shop resource allocation problem is to quickly find a mathematically optimal solution that the simulation model can test under conditions of variability. The model is made in a way that it can be altered or customized to a certain extent to test the system under a variety of situations. The model is a pure integer linear program having a single objective function and multiple constraints to limit the size of the feasible region.

Tables 3-1, 3-2 and 3-3 show the index sets, variable names and data sets associated with the integer program. These values have been changed from the actual program that was used and is shown in Appendix A for convenience and legibility.

Table 3-1: Index Sets for Integer Program

Index Sets	
i	Workstation
k	Product
t	Time period

Table 3-2: Variables in Integer Program

Variables	
x_{it}	Machines assigned to workstation i during time period t
y_{kt}	Unfinished products during time period t

Table 3-3: Data Sets for Integer Program

Data Sets	
p_{it}	Purchase cost of machine i during time period t
w_i	Original number of machines in workstation i
o_{it}	Operating cost of machine i during time period t
u_{kt}	Cost of each unfinished product during time period t
d_{kt}	Demand for product k during time period t
h_{ik}	Hours of processing of product k in workstation i
a_i	Hours of available processing time for workstation i
T	Total number of time periods
z	Target utilization of workstations

3.4.1 Objective Function

(1) *Minimize*

$$\sum_{t=1}^{1}\sum_{i}(p_{it}\times(x_{it}-w_i))+\sum_{t=2}^{T}\sum_{i}(p_{it}\times(x_{it}-x_{i,t-1}))$$
$$+\sum_{t=1}^{T}\sum_{i}\left(o_{i,1}\times x_{i,1}+o_{i,2}\times\left(x_{i,2}-x_{i,1}\right)+o_{i,3}\times\left(x_{i,3}-x_{i,2}\right)+\ldots+o_{i,T}\times\left(x_{i,T}-x_{i,T-1}\right)\right)$$
$$+\sum_{t}\sum_{k}\left(u_{kt}*y_{kt}\right)$$

The objective function for the integer program seeks to find the minimum total cost for T time periods. The total cost is made up of three components – machine purchase cost, machine operating cost and unfinished product cost. The program does this by finding the optimal number of machines to allocate to each workstation in each time period and the optimal number of products of each type to leave unfinished. The first parts are written in standard, linear form. It first sums the purchase cost of the first time period by multiplying the purchasing cost, *p*, by the difference in the number of machines between period 1 and the original state of the system. It then sums the purchase cost for the rest of the time periods after period 1. Next, the operating cost for each machine in each time period is added to the objective value. This formulation of the operating cost component allows for additional functionality for the decision makers. The time periods in this summation refer to the number of time periods the machine has been in the system. If operating or maintenance costs increase through the machine's life, this increase can be factored into the total cost as well. Finally, the last component of the objective function is a penalty function used to sum the cost of unfinished products at the end of each time period.

3.4.2 Constraints

The following constraints limit the size of the feasible region. The objective function then creates the gradient to find the optimal solution for the job shop. The first constraint of the formulation is the demand constraint:

$$(2) \quad \begin{cases} z \times (a_i \times x_{it}) + \sum_k (h_{ik} \times y_{kt}) > \sum_k (h_{ik} \times d_{kt}), \quad t = 1, \forall i \\ z \times (a_i \times x_{it}) + \sum_k (h_{ik} \times y_{kt}) > \sum_k (h_{ik} \times (d_{kt} + y_{k,t-1})), \quad t > 1, \forall i \end{cases}$$

The above constraint's purpose is to ensure that the demand for each product type is met or noted as unfinished product during each time period. The demand is calculated in terms of hours to more accurately determine the number of goods that should be left unfinished of each product type. In all periods except the first, the number of unfinished goods from the previous period is added to the total demand that the system must meet. Products that have not started to be processed or that are not completed are considered unfinished, and it is assumed that they are the first to finish processing during the successive time periods. Because of this, the performance measure of unfinished goods looks only at the final time period's number of unfinished goods. The left-hand side of the constraint contains the *x* and *y* variables and must be greater than the total demand to satisfy the constraint. This side sums the utilization, *z*, times the number of hours available with *x* number of machines for each workstation with the total number of hours of work that would be associated with *y* unfinished products. The proportion of *z* of the hours available from the workstation is used because it is the objective utilization for the workstations.

$$(3) \quad \begin{cases} x_{it} \geq w_i, & t = 1, \forall i \\ x_{it} \geq x_{i,t-1}, & t > 1, \forall i \end{cases}$$

Constraint (3) is a conditional constraint that controls the number of machines that are added to each workstation during each time period. During period 1, the number of machines must be greater than or equal to *w*, the number of machines originally in the system. For all other time periods, the number of machines must be greater than or equal to the number of machines in that workstation during the previous time period. This model of the job shop production ramp-up does not allow or test for decreases in machines of a certain workstation from one time period to the next. It is assumed that the job shop is expanding or at least keeping the number of machines in a given workstation steady when undergoing a time period transition.

$$(4) \quad \begin{matrix} x_{it} \geq 0 \\ y_{kt} \geq 0 \end{matrix} \quad \forall i, k, t$$

Constraint (4) is a non-negativity constraint that limits the variables to positive numbers. The total number machines and unfinished products must be non-negative for all workstations, products and time periods.

(5) x_{it}, y_{kt} integer $\forall i, k, t$

Constraint (5) ensures that the number of machines and unfinished products are integers, since fractional numbers are illogical. Constraints (4) and (5) together ensure that the variables are of the correct number type and set, i.e., non-negative integers.

3.4.3 Solution

The solution of the integer program, made up of the objective function and subsequent constraints, will provide the optimal number of machines needed in each time period and the number of products that should be left unfinished through the ramp-up period for the minimum cost ramp-up process. This solution is the optimal under a deterministic environment, which may not be the most accurate for a system which inherently has variability. The solution is sent to the simulation model as a starting point in testing under conditions of variability.

3.5 Simulation Model

The purpose of the simulation model is to test the robustness of the solution provided by mathematical modeling. The math model does not account for the routing of products through the system. It determines the number of machines to assign to each workstation based solely on the total number of hours of processing required. More importantly, it does not show how the system behaves under uncertainty and variability in processing times. Running the simulation allows the user to see which workstations are more highly utilized than others and where large queues build up. Targeting these areas can improve the system and lead to a superior production solution.

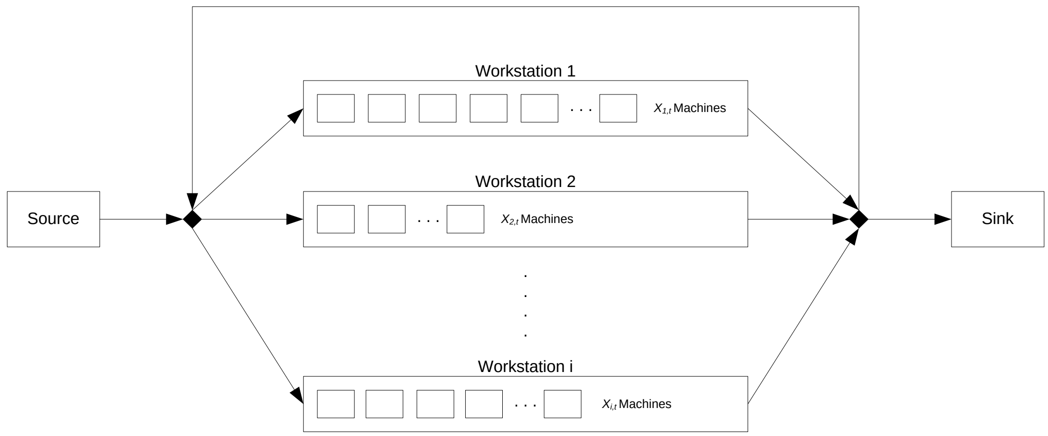


Figure 3-3: Representation of Simulation Model Structure

There are many different methods of producing a simulation model that could adequately represent a job shop, but they will all have some of the same general components. Figure 3-3 gives a simplified look at the model's general structure. First, there will need to be different types of model entities consistent with the number of product types, *k*. These entities are created at one or more sources according to an arrival rate consistent with the demand for the time period and are routed into the production system. The main part of the simulation model is a network of servers and paths that connect them. Each workstation can be represented by either a set of *x* servers or a single server with capacity equal to *x*. Each server has a processing time associated with it and each type of entity. The *i* workstations should be arranged in a manner that reflects the layout of the actual system represented. The entities are routed through the network according to their product type and leave the system through a sink when complete. After each of the *t* time periods, an event must occur to change the number of servers, *x*, and also the arrival rate of entities at the server according to the new time period's demand. In order to compare the different ramp-up strategies, statistics must be kept throughout the process during each time period. Further customization can also help make the model more representative of the actual system it represents.

This chapter has discussed the methodology behind the system that will determine the best possible ramp-up strategy. In the following chapter, these ideas and models are implemented into a real case study to test how they function in a job shop environment. A data set is applied to the optimization model, and a well known job shop model is used to test how the solution algorithm performs.

Chapter 4

Experimentation and Analysis

4.1 Algorithm

This chapter will serve to experiment with the production ramp-up system and show its effectiveness in solving a case study. The algorithm that was discussed in the previous chapter is used to reach a solution that meets a set of performance goals. The solution procedure will then be tested to show how sensitive it is to changes in the data using marginal analysis.

The first step of the algorithm is to enter the data set that reflects the job shop being tested. These data are displayed and explained in the following section. The statistics collected during the simulation runs are compared to the performance goals. For this case study, the performance goals that will be used are listed in Table 4-1.

Table 4-1: Performance Measures for Experiment

Performance Measure	Goal
Total Cost	< $1,500,000
Unfinished Products	< 1%
Workstation Utilization	< 80%
Average Number in Queue	< 2 Products
Average Time in System	< 1.5 * Base Process Time
Product 1	< 3.675 hours
Product 2	< 3.975 hours
Product 3	< 6.075 hours

If the system does not meet these performance goals, including the ramp-up project budget of $1,500,000, adjustments to the number of machines will have to be made. The simulation is run again with the new machine configuration and compared to the goal again. This sequence continues until all performance goals are met. The new solution may not be the least costly or have the fewest unfinished products, but it is a trade-off that decision makers must evaluate and compare to the stringency of their performance goals. Once all of the performance goals are met or deemed to be satisfactory, the resulting machine configuration is the best ramp-up strategy for the job shop system being analyzed.

4.2 Data

The inputs to the ramp-up system are important and need to be carefully determined so that the output will be useful to the user. If the data are not accurate, the system may produce an unexpected result or a production ramp-up plan that is irrational. Also, if the data do not resemble data that may be seen in a real production environment, it will have no use to the user or any job shop manager.

The job shop used in this case study closely follows the job shop model created by Law and Kelton (2000), and much of the data and structure of the job shop prior to its ramp-up phase is employed. It will experience a ramp-up phase consisting of five time periods (T) of equal length. The job shop will also have five workstations that process three distinct product lines. The following data are the inputs used in the system to demonstrate how it functions. These data sets follow those explained in Table 3-3 in the previous chapter.

Table 4-2: Purchase Cost of Machines by Workstation in each Time Period

Purchase Cost (p_{it})		Time Period				
		1	2	3	4	5
Workstation	1	$ 10,000	$ 10,000	$ 10,000	$ 10,000	$ 10,000
	2	$ 10,000	$ 10,000	$ 10,000	$ 10,000	$ 10,000
	3	$ 10,000	$ 10,000	$ 10,000	$ 10,000	$ 10,000
	4	$ 10,000	$ 10,000	$ 10,000	$ 10,000	$ 10,000
	5	$ 10,000	$ 10,000	$ 10,000	$ 10,000	$ 10,000

Table 4-2 describes the purchase cost of each additional machine during each of the 5 time periods. These numbers were generated to represent a substantial investment for the job shop. The purchase price of machines remains constant throughout all time periods because the time value of money principle is not taken into account. Time value of money could be applied using this system and is important in making capital expenditure decisions over larger time horizons. As stated in the assumptions, machines in each workstation are assumed to be identical regardless of what time period they are added to the system.

Table 4-3: Original Number of Machines in each Workstation

Original (w_i)		Machines
Workstation	1	3
	2	2
	3	4
	4	3
	5	1

Table 4-3 shows the number of machines that are in the job shop before any ramp-up has begun. The number of machines added to the system starting in period 1 adjoins to this amount. It is the basis for the job shop and is used in the warm-up period to initialize the simulation model. This initial machine configuration was established using the Law and Kelton (2000) model.

Table 4-4: Operating Cost of Machines by Workstation in each Time Period

Operating Cost (o_{it})		Time Period				
		1	2	3	4	5
Workstation	1	$ 100	$ 200	$ 300	$ 400	$ 500
	2	$ 100	$ 200	$ 300	$ 400	$ 500
	3	$ 100	$ 200	$ 300	$ 400	$ 500
	4	$ 100	$ 200	$ 300	$ 400	$ 500
	5	$ 100	$ 200	$ 300	$ 400	$ 500

Table 4-4 shows the operating cost of each machine during each time period of its lifetime. In the example that is used, the operating cost through the time periods increases steadily through the life of the machine. The increase is to reflect increased maintenance costs on machines through their lifetime and to make the system act more realistically. The machines that were originally in the system during the warm-up period are assumed to be in good condition and are assigned the first year's operating cost once the ramp-up process begins. This example will assume that the increases in operating and maintenance costs are equal for all types of machines in the workstations. Like the purchase price, the reaction to changes in operating cost is explored using sensitivity analysis.

Table 4-5: Cost of Unfinished Products in each Time Period

Unfinished Cost (u_{kt})		Time Period				
		1	2	3	4	5
Product	1	$ 150	$ 150	$ 150	$ 150	$ 150
	2	$ 150	$ 150	$ 150	$ 150	$ 150
	3	$ 200	$ 200	$ 200	$ 200	$ 200

Unfinished cost represents the cost assumed per product that was not completed during the desired time period. The remaining products that are not finished are added to the demand for the subsequent time period, and it is assumed that they are finished before work begins on the next time period's demand. As shown in Table 4-5, the unfinished cost for products 1 and 2 is $150 each, and the cost for product 3 is $200 to reflect the extra processing time it requires that adds additional value. Table 4-7 shows the total base processing time for each product, and product 3 clearly requires significantly more processing than the other product types. Like the other cost data sets, the sensitivity of the unfinished cost is explored using marginal analysis. It is difficult to determine an exact cost of unfinished products, so an abstract number is formulated. The selling price of the product is a part of the unfinished cost price, but also non-quantifiable aspects such as loss of future business and goodwill must be factored into the cost as well.

Table 4-6: Demand for each Product Type by Time Period

Demand (d_{kt})		**Time Period**				
		1	2	3	4	5
Product	1	600	1200	1800	2400	3000
	2	1000	2000	3000	4000	5000
	3	400	800	1200	1600	2000
	Total	**2000**	**4000**	**6000**	**8000**	**10000**

Table 4-6 represents the total demand that the job shop must meet throughout the time periods of the production ramp-up. It is divided up by product for the mathematical model to read the data, but the totals are given at the bottom, and these are what are used to look at the overall ramp-up of the system. These values are chosen to look at a linear ramp-up approach, but other ramp-up plans, such as a more exponential approach, are explored in the marginal analysis section. The product mix was set by the Law and Kelton model (2000) and consisted of 30% Product 1, 50% Product 2 and 20% Product 3. These proportions are kept constant throughout all 5 time periods and the initial warm-up time period. The warm-up period’s demand was also set by Law and Kelton who assumed an interarrival rate of 0.25 hours, which is equivalent to 960 units in the 240 hours for which the job shop operates each time period. In the simulation model, the same amount of demand is used, but these values are transferred into interarrival times according to the 240 hours that the job shop operates per time period. The interarrival rate was constant to simulate a realistic flow of a known demand for each time period.

Table 4-7: Processing Time (hours) Required for each Product Type at each Workstation

Hours (h_{ik})		Product		
		1	2	3
Workstation	1	0.6	0.8	0.7
	2	0.85	-	1.2
	3	0.5	0.75	1
	4	-	1.1	0.9
	5	0.5	-	0.25
	Total	**2.45**	**2.65**	**4.05**

Table 4-7 lists the amount of time (in hours) that each product type spends at each workstation as it is routed through the system. These numbers were also adopted from the Law and Kelton model (2000). The total number of hours that each product spends in the entire system is summed at the bottom of the table. In the simulation model, the actual time that each product spends in processing at each workstation is generated using a normal distribution with mean of the times listed and a standard deviation of 10% of the mean. As can be seen in Table 4-7, not all products are processed in every workstation, and Product 3 has a significantly higher total processing time than the other two product lines.

Table 4-8: Number of Available Hours at each Machine in Workstation *i*

Available (a_i)		Hours
Workstation	1	240
	2	240
	3	240
	4	240
	5	240

Table 4-8 shows the number of hours that each machine in each workstation can devote to producing the three product lines during one time period. This number can easily be altered in order to change the work mix in the shop or to instantly change the utilization of shop hours. The number of hours that was chosen, 240, assumes that the job shop operates for one 8 hour shift per day for 30 days a month. One month transitions were used for the time periods, but larger ramp-up processes may require longer time intervals with more demand and more available hours. These data points can easily be changed in the ramp-up system for easy customization.

Utilization is clearly an important performance metric for a job shop environment with increasing throughput. It allows the job shop managers to gauge machine performance and bottlenecks in the job shop. The system allows for users to input their own objective utilization value, but a value of 0.8 was used for this case study. The number was derived from Hopp and Spearman (2008) who developed Figure 4-1. This utilization value is factored into the optimization program to quickly approximate the number of machines in each workstation required in order to achieve the utilization goal.

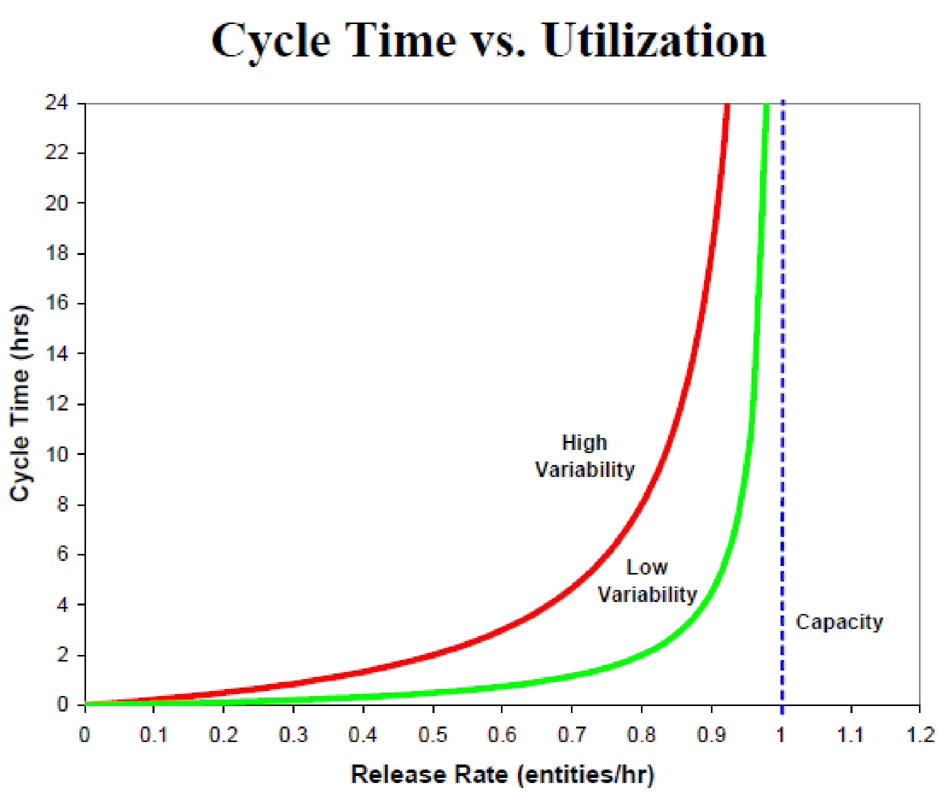


Figure 4-1: Cycle Time vs. Utilization (adopted from Hopp and Spearman, 2008)

Figure 4-1 shows the relationship between utilization and cycle time for both high and low variability systems. As the system's utilization reaches about 80%, the cycle time for each part begins to rise drastically. The system presented has relatively little variability, so 80% was chosen as the upper limit for workstation utilization. This constraint is repeated for all *i* workstations and *t* time periods.

Table 4-9: Number of Unfinished Products (Data after Iteration 1)

Unfinished Products (y_{kt})		Time Period				
		1	2	3	4	5
Product	1	6	13	21	23	34
	2	12	23	31	45	54
	3	6	13	19	28	37
	Total	24	49	71	96	125

Though the number of unfinished products, y_{kt}, is a variable, the user has an option of fixing the parameter to specific values as in Table 4-9. The main purpose for this functionality is to relay the simulation solution back to the optimization model when the user is trying to converge on an optimal solution. During the first iteration, the number of unfinished products is a variable, but in the subsequent iterations, it becomes data and an input into the optimization model. These values are pulled directly from statistics generated during the job shop simulation.

4.3 Optimization Model

$$(1)\quad \textit{Minimize}\quad \sum_{t=1}^{1}\sum_{i=1}^{5}(p_{it}\times(x_{it}-w_i))+\sum_{t=2}^{5}\sum_{i=1}^{5}(p_{it}\times(x_{it}-x_{i,t-1}))$$
$$+\sum_{t=1}^{1}\sum_{i=1}^{5}\left(o_{it}\times x_{it}\right)$$
$$+\sum_{t=2}^{2}\sum_{i=1}^{5}\left(o_{i,t-1}\times\left(x_{it}-x_{i,t-1}\right)+o_{it}\times x_{i,t-1}\right)$$
$$+\sum_{t=3}^{3}\sum_{i=1}^{5}\left(o_{i,t-2}\times\left(x_{it}-x_{i,t-1}\right)+o_{i,t-1}\times\left(x_{i,t-1}-x_{i,t-2}\right)+o_{it}\times x_{i,t-2}\right)$$
$$+\sum_{t=4}^{4}\sum_{i=1}^{5}\left(o_{i,t-3}\times\left(x_{it}-x_{i,t-1}\right)+o_{i,t-2}\times\left(x_{i,t-1}-x_{i,t-2}\right)+o_{i,t-2}\times\left(x_{i,t-2}-x_{i,t-3}\right)+o_{it}\times x_{i,t-3}\right)$$
$$+\sum_{t=5}^{5}\sum_{i=1}^{5}\left(o_{i,t-4}\times\left(x_{it}-x_{i,t-1}\right)+o_{i,t-3}\times\left(x_{i,t-1}-x_{i,t-2}\right)+o_{i,t-2}\times\left(x_{i,t-2}-x_{i,t-3}\right)+o_{i,t-2}\times\left(x_{i,t-3}-x_{i,t-4}\right)+o_{it}\times x_{i,t-4}\right)$$
$$+\sum_{t=1}^{5}\sum_{k=1}^{3}\left(u_{kt}*y_{kt}\right)$$

$$(2)\quad \begin{cases} 0.8\times\left(a_i\times x_{it}\right)+\sum_{k=1}^{3}\left(h_{ik}\times y_{kt}\right)>\sum_{k=1}^{3}\left(h_{ik}\times d_{kt}\right), & t=1,\forall i \\ 0.8\times\left(a_i\times x_{it}\right)+\sum_{k=1}^{3}\left(h_{ik}\times y_{kt}\right)>\sum_{k=1}^{3}\left(h_{ik}\times\left(d_{kt}+y_{k,t-1}\right)\right), & t>1,\forall i \end{cases}$$

$$(3)\quad \begin{cases} x_{it}\geq w_i, & t=1,\forall i \\ x_{it}\geq x_{i,t-1}, & t>1,\forall i \end{cases}$$

$$(4)\quad \begin{matrix} x_{it}\geq 0 \\ y_{kt}\geq 0 \end{matrix}\quad \forall i,k,t$$

$$(5)\quad x_{it},y_{kt}\ \text{integer}\quad \forall i,k,t$$

The data described in the previous section are used to run the optimization model. The complete analytical model is shown above with the objective function expanded and some of the data filled in. Lingo optimization software 13.0 was used to solve the mathematical program. The full model’s code can be seen in Appendix A, and the model, as generated by Lingo, is shown in Appendix B. Lingo was chosen because it provides an easy-to-use environment to quickly write many constraints of the same type. It also functions seamlessly with Excel and other programs to make the interface easier to use and build. The unique language uses sum loops and for loops to easily sum many variables or generate many constraints using a single line of code. The Lingo

Solver uses the branch and bound technique and generally solves in one second or less. The same model could be solved, however, in many different optimization languages.

Solving the optimization model described above provides the following results. The total number of machines in the job shop per workstation as well as the number of products of each type that are left unfinished are shown in Table 4-10. At the bottom, the total cost of the ramp-up strategy is broken down to show where the costs are being incurred. This initial solution will next be assessed using the simulation model.

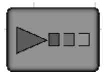
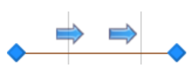
Table 4-10: Optimization Model Results (Iteration 1)

Number of Machines		Time Period						
		0	1	2	3	4	5	
Workstation	1	3	8	15	23	30	36	
	2	2	6	11	16	21	24	
	3	4	8	16	23	31	36	
	4	3	8	16	23	31	36	
	5	1	3	5	7	9	10	
Unfinished Products								
Product	1		0	0	0	0	25	
	2		0	0	0	0	132	
	3		0	0	0	0	270	
Subtotal:			0	0	0	0	427	
Total:							**427**	
Machine Acquisition Cost:			$ 200,000.00	$ 300,000.00	$ 290,000.00	$ 300,000.00	$ 200,000.00	$ 1,290,000.00
Operating Cost:			$ 3,300.00	$ 9,600.00	$ 18,800.00	$ 31,000.00	$ 45,200.00	$ 107,900.00
Unfinished Cost:			$ -	$ -	$ -	$ -	$ 77,550.00	$ 77,550.00
Total Cost:			$ 203,300.00	$ 309,600.00	$ 308,800.00	$ 331,000.00	$ 322,750.00	**$ 1,475,450.00**

4.4 Simulation Model

The other component of the production ramp-up system is the simulation model. Table 4-11 explains the elements used in the simulation model and how they are utilized specifically in the model. The full simulation, which utilizes these elements, can be seen in Figure 4-2. The simulation of the job shop environment to test the capacity ramp-up plan is created using the Simio simulation software. Simio is an object-oriented, multi-paradigm simulation language that allows for a wide array of applications with wide flexibility. Simio provides a highly visible environment for simulating the job shop and flexible process capabilities, which made collecting statistics and performance measures simple. Each simulation run was replicated 10 times, and the statistics were averaged over these 10 runs to ensure the significance of the results.

Table 4-11: Elements of the Simulation Model

Element	Simio Symbol	General Information	Simulation Constraints
Source		A source object creates entities at a specified rate and pattern. It has an output buffer where entities can be seized before leaving the source.	The source generates entities at a rate which changes each time period based on that period's demand. It creates entities based on their product mix and assigns the appropriate routing sequence before the entity leaves the source.
Sink		A sink object destroys entities and records statistics for the simulation. It has an input buffer where entities can be seized before they are destroyed.	The sink destroys entities to clear up system memory. In the simulation, it is used to tally the total amount of time in system for each product for calculating the average time in system statistic.
Server		A server object has three queues for processing entities. It has input and output buffers to store entities before and after they are processed. It also has a processing station queue with a specified capacity where objects are processed for a specific or variable amount of time.	There are five servers in the simulation model, and each of them is assigned its respective capacity based on a data table which lists all of the capacities for all of the servers in each of the five time periods. The model entities are processed at the servers for a designated amount of time following a normal distribution. The three different product lines are routed through a sequence of servers in different orders and are processed for a range of time intervals.
Path		A path object is a link between two nodes. Travel time is determined based on the length of the path and the traveler's speed.	Paths are used to connect the source, servers and sink. The model entities, or products, use the paths to be routed through the system.
Model Entities		Model entities are objects that can have intelligent behavior and make decisions as they travel through a model. They travel along paths through the system and are processed by the other model objects.	The model entities in the simulation represent the products. Three independent model entities are used, one for each type of product. They are given a sequence to follow through the system based on their type of model entity. These model entities are used to simulate flow of products through the job shop.
Timer		A timer is an element used to fire one or more events.	Timer elements were used to change the system's properties at the end of each time period. Six timers were needed, one for each time period, including the warm-up period. The timers fire events which change the capacity of the servers at the end of each time period as well as collect statistics. These statistics include the number created, number destroyed, server utilization, average number in queue and average time in system.

4.4.1 Structure and Layout

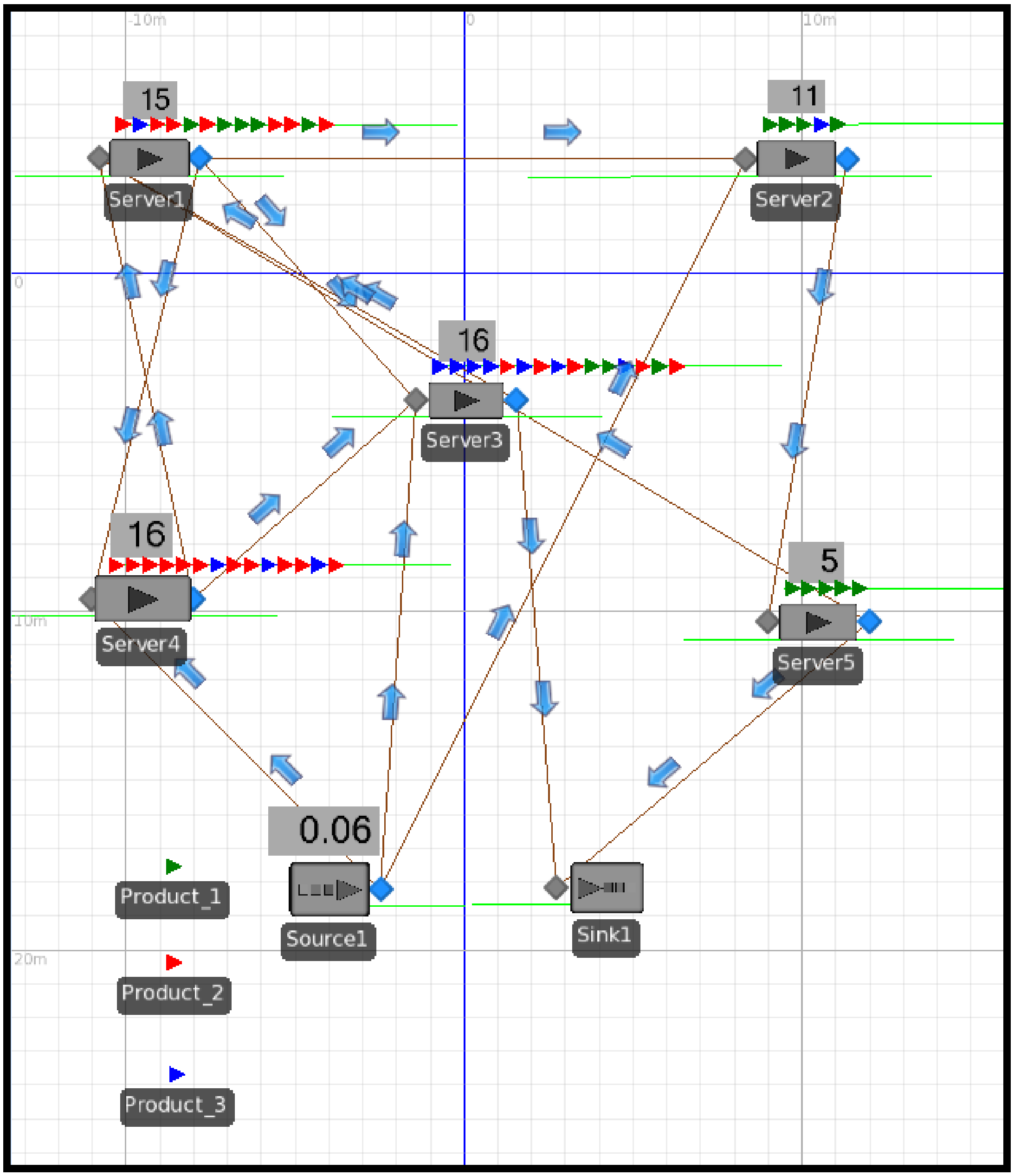


Figure 4-2: Structure of the Simulation Model (Simio)

The simulation model of the job shop environment is built using a relatively simple set of components. It consists of a source, sink and five servers that represent the five different workstations. The source creates one of three different model entities that represent the three separate product lines produced in the job shop. The entities are created using a constant interarrival time based on the demand in each time period. Since the demand for each period was forecast, it is assumed that the products can be introduced to the system in a fairly constant manner. The entities are assigned a product type based on their probabilistic weights of 0.3, 0.5 and 0.2 respectively. The type of product determines its routing and processing time through the system of workstations as is summarized in Table 4-12.

Table 4-12: Simulation Product Mix and Routing

Product	Product Mix	Routing (Workstations)
1	0.3	3,1,2,5
2	0.5	4,1,3
3	0.2	2,5,1,4,3

The servers begin with an initial capacity, or number of machines, and the increase in machines through the time periods is represented by corresponding increases in capacity. The server processing times are product-specific, so they are assigned according to the sequence tables that route the different products through the system. These processing times were discussed previously in Table 4-7. They were those used in the Law and Kelton (2000) model. They follow a normal distribution with a coefficient of variation of 0.1 for each of the processing times. This is a relatively low amount of variation in the processing times, which can be assumed for a process with such large throughput. Finally, after the entities are processed through their respective sequence of workstations, they are sent to the sink to be destroyed and removed from the system.

4.4.2 Time Period Transitions

It is assumed that the job shop runs for one 8-hour shift, 7 days a week, or 240 hours per month. There is also no delay or productivity change during individual days or between days. It is assumed that work is stopped at the end of a shift and picked up where it was left off the next day. This allows the simulation to run continuously for 240 hours of simulation time. Each month is considered a new time period where the capacity of the workstations is changed based on the results from the mathematical program. The simulation model recognizes the change in time periods based on a set of events triggered to go off after each 240 hour simulation period. These events also allow the simulation to gather period-specific statistics.

4.4.3 Performance Measures

The performance measures that are gathered using Simio's process capabilities include workstation utilization, average number in queue and average time in system by product. These statistics along with the total cost of the ramp-up and the number of unfinished products are used to determine the quality of the solution. The number of products created and destroyed during each period is collected in order to verify that the model is producing and destroying entities correctly as well. These data points were verified using Simio's experiment capabilities on individual time periods as well, but processes were used in order to collect the statistics using a single simulation run. Utilization, average number in queue and average time in system were chosen as performance measures for the system because they can provide a wide array of information about bottlenecks and how workstations and different product types are behaving. Other performance measures can be calculated or estimated using these figures, but many provide redundant information.

A summary of the results from running the integer program solution of iteration 1 through the simulation model are shown in Table 4-13. These results show the key performance measures and allow the decision makers to compare them to their goals, in this case, those listed in Table 4-1.

Table 4-13: Simulation Model Results (Iteration 1)

Number of Machines

Period	Workstation 1	Workstation 2	Workstation 3	Workstation 4	Workstation 5
0	3	2	4	3	1
1	8	6	8	8	3
2	15	11	16	16	5
3	23	16	23	23	7
4	30	21	31	31	9
5	36	24	36	36	10

Performance Measures	Result
Unit Cost	$ 48.54
Total Cost	$ 1,456,150.00
Unfinished Products	124.1
% of Total	0.41%

Utilization (%)

Period	Workstation 1	Workstation 2	Workstation 3	Workstation 4	Workstation 5
0	93.99	97.68	70.62	94.27	78.55
1	75.23	69.82	75.70	76.32	56.41
2	79.75	75.22	75.37	75.96	66.35
3	78.19	76.90	78.65	79.36	70.86
4	79.91	78.18	77.76	78.57	73.59
5	83.19	86.32	83.76	84.13	83.63

Average Number in Queue

Period	Workstation 1	Workstation 2	Workstation 3	Workstation 4	Workstation 5
0	1.55	10.91	0.14	3.56	0.27
1	0.11	0.48	0.22	0.27	0.08
2	0.14	0.25	0.11	0.10	0.13
3	0.06	0.22	0.14	0.15	0.18
4	0.07	0.21	0.08	0.07	0.22
5	0.11	0.67	0.22	0.20	0.54

Average Time in System (Hours)

Period	Product 1	Product 2	Product 3
0	8.35	4.32	11.19
1	2.74	2.76	4.50
2	2.52	2.68	4.14
3	2.50	2.67	4.12
4	2.49	2.66	4.10
5	2.53	2.67	4.14

Comparing these results to the performance goals, it can be seen that the only goal that hasn't been achieved in iteration 1 is the utilization of less than 80%. It is violated in all five workstations during time period 5. This signals the user to proceed to iteration 2.

4.5 Solution Procedure

The design of the production ramp-up system seeks to find a solution in which the optimization and simulation models converge. This solution is optimal under the conditions of variability that the simulation model introduces to the system. If this solution does not meet all of the criteria set forth by the decision makers, they would be able to marginally adjust the solution until they are satisfied with the performance metrics. All criteria may not be satisfied since they are inversely related (i.e., utilization vs. average time in queue), and the decision makers may have to prioritize the criteria or assign heavier weight to those deemed most important.

After the initial iteration, each of the following iterations is slightly different in that the number of unfinished goods is no longer a decision variable. The number of unfinished goods is then pulled from the simulation model and becomes data in the following iteration's optimization model. When the optimization model is run again using the number of unfinished products from iteration 1, the following results are attained. Note that the unfinished goods figures shown are an input, not an output, to this model.

Table 4-14: Optimization Model Results (Iteration 2)

Number of Machines		Time Period						
		0	1	2	3	4	5	
Workstation	1	3	8	15	23	30	38	
	2	2	6	11	16	21	26	
	3	4	8	16	23	31	38	
	4	3	8	16	23	31	38	
	5	1	3	5	7	9	11	
Unfinished Products								
Product	1		6	13	21	23	34	
	2		12	23	31	45	54	
	3		6	13	19	28	37	
Subtotal:			24	49	71	96	125	
Total:							**365**	
Machine Acquisition Cost:			$ 200,000.00	$ 300,000.00	$ 290,000.00	$ 300,000.00	$ 290,000.00	$ 1,380,000.00
Operating Cost:			$ 3,300.00	$ 9,600.00	$ 18,800.00	$ 31,000.00	$ 46,100.00	$ 108,800.00
Unfinished Cost:			$ 3,900.00	$ 8,000.00	$ 11,600.00	$ 15,800.00	$ 20,600.00	$ 59,900.00
Total Cost:			$ 207,200.00	$ 317,600.00	$ 320,400.00	$ 346,800.00	$ 356,700.00	**$ 1,548,700.00**

Compared to iteration 1, this solution adds two additional machines to workstations 1-4 and a single additional machine to workstation 5 in the final time period. It decreases the total unfinished cost for the job shop but increases the total cost of the ramp-up. Running this optimized solution through the simulation model yields the following results in Table 4-15.

Table 4-15: Simulation Model Results (Iteration 2)

Number of Machines

Period	Workstation 1	2	3	4	5
0	3	2	4	3	1
1	8	6	8	8	3
2	15	11	16	16	5
3	23	16	23	23	7
4	30	21	31	31	9
5	38	26	38	38	11

Performance Measures	Result
Unit Cost	$ 51.56
Total Cost	$ 1,546,900.00
Unfinished Products	122.9
% of Total	0.41%

Utilization (%)

Period	Workstation 1	2	3	4	5
0	93.99	97.68	70.62	94.27	78.55
1	75.23	69.82	75.70	76.32	56.41
2	79.75	75.22	75.37	75.96	66.35
3	78.19	76.90	78.65	79.36	70.86
4	79.91	78.18	77.76	78.57	73.59
5	78.80	79.63	79.33	79.74	76.01

Average Number in Queue

Period	Workstation 1	2	3	4	5
0	1.55	10.91	0.14	3.56	0.27
1	0.11	0.48	0.22	0.27	0.08
2	0.14	0.25	0.11	0.10	0.13
3	0.06	0.22	0.14	0.15	0.18
4	0.07	0.21	0.08	0.07	0.22
5	0.03	0.21	0.08	0.07	0.25

Average Time in System (Hours)

Period	Product 1	2	3
0	8.35	4.32	11.19
1	2.74	2.76	4.50
2	2.52	2.68	4.14
3	2.50	2.67	4.12
4	2.49	2.66	4.10
5	2.48	2.66	4.09

The simulation results support the suggestion of higher total cost for the ramp-up. Clearly, adding machines to the workstations decreases their total utilization but, at the same time, it increases both purchase and operating expenses. These trade-offs must be weighed by the decision makers, and the system allows for marginal adjustments until the solution is satisfactory.

In this case study, it is assumed that the target utilization was 80%, but it is not as critical to the ramp-up as total cost. The budget for the ramp-up project has been set at the goal of $1,500,000, and this cannot be exceeded. The utilization goal is less of a priority and can be sacrificed in order to meet the cost restriction. Since the utilization goal is going to be relaxed, machines will marginally be removed from the workstation with the lowest current utilization in the final time period. In this case, since workstation 5 has the lowest utilization of 76.01%, the number of machines in time period 5 should be decreased from 11 machines to 10. This solution is run through the simulation model to determine the new total cost and see if any other performance goals are then violated. This process continues until the solution is acceptable to the decision makers. The marginal adjustment iterations performed in this study can be seen in Appendix D. The final solution was achieved when one machine was removed from each workstation producing a total cost of $1,496,550. Full details of the solution are shown in Table 4-16. Each workstation's utilization is slightly over the goal of 80%, but this solution is deemed acceptable with the budget restriction placed on the ramp-up.

Table 4-16: Simulation Model Results (Final Iteration)

Number of Machines

Period	Workstation 1	2	3	4	5
0	3	2	4	3	1
1	8	6	8	8	3
2	15	11	16	16	5
3	23	16	23	23	7
4	30	21	31	31	9
5	37	25	37	37	10

Performance Measures	Result
Unit Cost	$ 49.89
Total Cost	$ 1,496,550.00
Unfinished Products	124.3
% of Total	0.41%

Utilization (%)

Period	Workstation 1	2	3	4	5
0	93.99	97.68	70.62	94.27	78.55
1	75.23	69.82	75.70	76.32	56.41
2	79.75	75.22	75.37	75.96	66.35
3	78.19	76.90	78.65	79.36	70.86
4	79.91	78.18	77.76	78.57	73.59
5	80.97	82.27	81.54	82.34	82.79

Average Number in Queue

Period	Workstation 1	2	3	4	5
0	1.55	10.91	0.14	3.56	0.27
1	0.11	0.48	0.22	0.27	0.08
2	0.14	0.25	0.11	0.10	0.13
3	0.06	0.22	0.14	0.15	0.18
4	0.07	0.21	0.08	0.07	0.22
5	0.06	0.32	0.14	0.13	0.59

Average Time in System (Hours)

Period	Product 1	2	3
0	8.35	4.32	11.19
1	2.74	2.76	4.50
2	2.52	2.68	4.14
3	2.50	2.67	4.12
4	2.49	2.66	4.10
5	2.51	2.67	4.11

Table 4-17: Summary of Results

	Iteration	Total Cost	Total Unfinished Goods	Average Utilization (Period 5)	Average Number in Queue (Period 5)	Average Time in System (Period 5)		
						Product 1	Product 2	Product 3
	1	$ 1,456,150.00	124.1	84.21	0.35	2.53	2.67	4.14
	2	$ 1,546,900.00	122.9	78.70	0.13	2.48	2.66	4.09
Marginal Analysis	3	$ 1,536,800.00	122.6	80.10	0.23	2.51	2.66	4.11
Marginal Analysis	4	$ 1,526,500.00	122.2	80.63	0.23	2.51	2.66	4.11
Marginal Analysis	5	$ 1,516,400.00	121.7	80.95	0.23	2.50	2.66	4.12
Marginal Analysis	6	$ 1,506,500.00	122.6	81.59	0.25	2.51	2.66	4.12
Marginal Analysis	7	$ 1,496,550.00	124.3	81.98	0.25	2.51	2.67	4.11

Table 4-17 provides a summary of the results throughout all the iterations of the solution process. Iterations 1 and 2 were those resulting from the optimization and simulation models converging on the results in iteration 2. The remainder of the iterations is the result of the marginal removal of machines in order to decrease the total cost to meet the budget restriction. Statistics here, including utilization, average number in queue and average time in system, are only shown for period 5, since it was the only period in which they exceeded the limits set for the ramp-up. Marginal analysis shows to have brought down total cost by roughly $50,000, while increasing utilization by about 3% on average to slightly exceed the limit. If this trade-off is deemed acceptable, then this is the final production ramp-up solution.

4.6 Sensitivity Analysis

Sensitivity analysis is necessary for this type of system in order to determine the effects of changes to the system inputs on the resulting machine outputs and total cost of the system. Specific changes to the system or the data that drives it can have a large impact on the resulting output, and this information is critical to fully understand the system and the decision support tool. Some changes or trials may be unrealistic in the actual job shop environment, but they would provide a good idea of the robustness of the system from the data standpoint. Total cost will be the main indicator of the performance of the system under marginal changes. One-factor-at-a-time (OFAT) experiments will be run on several different input parameters to demonstrate the reaction and its magnitude to the system as a whole. These changes and the different trials that will be performed will be explained, but testing solely one independent variable at a time will provide a good idea of how the variable makes the system react.

4.6.1 Demand

Demand is the basis for the production ramp-up, and it has the single largest impact on cost and production for the system. It determines how the job shop ramp-up process will occur and whether the ramp-up follows one of three different ramp-up strategies. The system can experience a linear ramp-up in which the demand for products is increased by an equal amount in each time period. It can also undergo a ramp-up strategy that is weighted toward the beginning or end of the ramp-up period. A ramp-up strategy that is more heavily weighted in the beginning with larger demand increases in the earlier time periods will be referred to as following a logarithmic pattern, while one with larger demand increases toward the end of the ramp-up follows an exponential ramp-up strategy. There are varying degrees of growth rates for linear, logarithmic and exponential ramp-up strategies, and several are explored here. Figure 4-1 shows five strategies: logarithmic with rapid growth, logarithmic with slow growth, linear, exponential with slow growth and exponential with rapid growth. The figure plots the demand in number of products for each time period, including the warm-up period in which 960 units are produced. Exploring these ramp-up strategies provides a strong understanding of how a variety of strategies would perform.

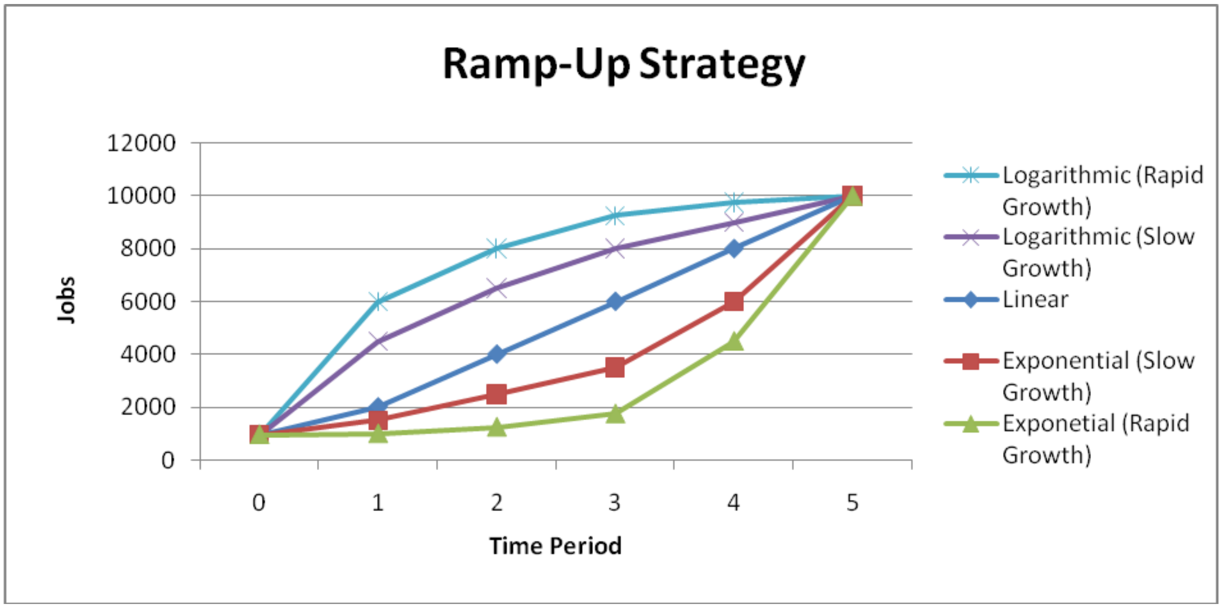


Figure 4-3: Ramp-Up Strategies Tested Using Marginal Analysis

Each of the ramp-up strategies presented has its own strengths and weaknesses. Some may be better suited for specific products or manufacturing environments, or the necessary strategy may be determined by the demand for the product. Oftentimes, technological products and electronics have large demand in the beginning of their lifecycle, but it dissipates as the technology become outdated or obsolete. In cases such as this, it is advantageous for the manufacturer to build as many products in the shortest time frame possible. Table 4-18 breaks down the cost for each ramp-up strategy and provides some interesting results. The strategies with faster ramp-up rates in the earlier time periods have higher total cost, primarily due to operating costs, but have lower unit costs for their products. Faster ramp-up rates mean that more products will be made throughout the ramp-up process, so this drives down the unit costs of products.

Table 4-18: Marginal Analysis Results for Ramp-Up Strategies

Strategy	Price per Time Period ($)	Total Cost	Finished Products	Unit Cost
Logarithmic (Rapid)	6000, 8000, 9250, 9750, 10000	$ 1,646,750.00	42,879.4	$ 38.40
Logarithmic (Slow)	4500, 6500, 8000, 9000, 10000	$ 1,518,200.00	37,876.9	$ 40.08
Linear	2000, 4000, 6000, 8000, 10000	$ 1,457,800.00	29,875.9	$ 48.80
Exponential (Slow)	1500, 2500, 3500, 6000, 10000	$ 1,334,150.00	22,485.7	$ 59.33
Exponential (Rapid)	1000, 1250, 1750, 4500, 10000	$ 1,360,450.00	16,102.3	$ 84.49

Determining which strategy is best for a particular job shop or product line may depend on demand for the product, or it may be constrained by funding for the ramp-up process. As it can be seen in the logarithmic ramp-up strategies, a larger initial investment is necessary in order to satisfy the larger demand in early time periods. Clearly, these types of ramp-ups yield superior solutions when measured by unit price, but such early, drastic changes to the manufacturing system could produce unprofitable results. Terwiesch and Bohn (2001) explored learning and process improvement during production ramp-up and concluded that it is optimal for a system to perform more experimentation for learning purposes earlier in the ramp-up process and decrease the experimentation as the process continues until the system is operating at full capacity. It was determined that this was the best strategy even when product prices are at their highest at the beginning of the ramp-up process. These results favor a more exponential ramp-up strategy, despite the larger unit costs. One interesting feature about the exponential ramp-up strategy can be seen in the Exponential (Rapid) strategy. Once the ramp-up rate becomes steep enough, the system is unable to process the products as fast as they are being introduced. This creates large queues in the system and leaves more products unfinished. Care must be taken to avoid this type of scenario, as the unit cost of the products skyrockets.

4.6.2 Purchasing Cost

The total cost of the ramp-up process is made up of purchasing cost, operating cost and unfinished products cost. Purchasing cost of the machines for the workstations makes up the majority of the expenses in the ramp-up, so it is clearly an important area to analyze when optimizing the system and finding the minimum cost solution that is acceptable. Several pricing scenarios were explored to determine how the purchasing cost of the machines impacts the solution. These different pricing scenarios included prices ranging from $5,000 to $50,000. The summary of the resulting simulation costs and the number of unfinished products after the 5 time periods analyzed can be seen below in Table 4-19.

Table 4-19: Marginal Analysis Results for Machine Purchase Price

Price per Machine	Total Cost	Unfinished Products
$5,000	$ 858,250.00	122.9
$7,500	$ 1,203,250.00	122.9
$10,000	$ 1,457,800.00	124.1
$12,500	$ 1,557,600.00	149.4
$15,000	$ 1,820,350.00	152.0
$20,000	$ 2,325,000.00	243.6
$30,000	$ 3,170,000.00	1,797.4
$50,000	$ 4,743,000.00	4,110.2

As expected, the scenarios with larger prices have higher overall cost. Higher priced machines directly increase the objective function value and the total cost of the ramp-up. As the machine prices decrease, the job shop is less hesitant to purchase new machines to process more products. At higher machine prices, the job shop is more inclined to incur the unfinished cost, especially in later time periods. The number of unfinished goods remains relatively constant and less than 1% of the total demand until some point between the $20,000 and $30,000 purchase prices. Compared to the relatively low cost of unfinished goods at these high machine prices, the number of goods left unfinished is surprisingly low. The purchase cost clearly can have a large impact on the solution for the ramp-up process, but it is robust in a wide range of rational prices.

4.6.3 Operating Cost

Operating cost is another component of the total cost of the system, which makes up the next largest portion of the total after the purchasing cost. The operating cost works slightly differently than the purchasing cost table, as operating cost is measured retroactively as well as in the current time period. The older the machine, the more the operating cost for that period due to the need for increased maintenance. Generally, as operating costs increase, the total simulation cost increases. Table 4-20 summarizes the results of the 5 trials.

Table 4-20: Marginal Analysis Results for Operating Cost

Scenario	Price per Time Period ($)	Total Cost	Unfinished Products
Linear 1	50, 100, 150, 200, 250	$ 1,403,850.00	124.1
Linear 2	100, 200, 300, 400, 500	$ 1,457,800.00	124.1
Linear 3	200, 400, 600, 800, 1000	$ 1,455,100.00	134.5
Exponential 1	100, 200, 400, 800, 1600	$ 1,528,500.00	124.1
Exponential 2	1000, 2000, 4000, 8000, 16000	$ 2,914,850.00	145.5

Despite the increases in simulation costs, there were only slight changes in two of the experiments. The scenarios Linear 1, 2 and Exponential 1 produced machine configurations that were identical. The operating costs during the time periods were all that differed among the scenarios causing the difference in total costs. The Linear 3 scenario had fewer machines in each of the 5 workstations during the last time period. Its total cost actually decreased from the other linear scenarios, and this can be attributed to high workstation utilization values. The utilization values were also very high in the Exponential 2 scenario, and this helped to keep the total cost low for these scenarios and the number of unfinished goods similar throughout all of the scenarios. Exponential 2 showed the largest difference in machine configuration, and it still was not drastically different. It allocated the fewest machines to the workstations in the final time period and was the only scenario that altered the number of machines in any of the earlier time periods. Time period 1 showed one machine fewer than in each of the other scenarios in order to avoid the large penalty of $16,000 incurred during those machines' final year of operation. The operating costs had to be raised to the unrealistic levels shown for Exponential 2 in order for the system to produce a solution that was not similar to the others. The values assumed in the Exponential 2 scenario were extreme in that the operating cost eventually became more expensive than the purchase price of the machines. This experimentation showed that the system was robust in how it utilizes operating costs and that the distribution of the operating costs over time made little difference.

4.6.4 **Unfinished Cost**

Unfinished cost is the final and smallest component of the total ramp-up cost. Again, it is an obscure value that is made up of the lost revenue from the sale of the product and loss of goodwill. It is the most volatile and creates different solutions for relatively small changes. This is partially due to how the objective function was formulated, calculating the number of unfinished products using a penalty function. It is also partially due to the fact that the reduction of a single machine can drastically increase the average number of products in queue, thus increasing the number of products that are left unfinished. Table 4-21 summarizes the results of 5 experiments that were run using different unfinished cost values.

Table 4-21: Marginal Analysis Results for Product Unfinished Cost

Unfinished Cost	Total Cost	Unfinished Products
$25	$ 1,010,150.00	10,393.1
$50	$ 1,119,500.00	3,454.9
$100	$ 1,255,000.00	149.4
$150	$ 1,293,450.00	140.3
$200	$ 1,460,600.00	124.1

The total cost of the ramp-up process does not accurately reflect the difference in the solutions that are presented under these different scenarios. When the unfinished cost is assigned to be $25 throughout the process, over 10,000 products are left unfinished, which is roughly 35% of the total demand. Even at $50, almost 12% of the total demand was left unfinished. The optimization program found it less costly to assume this unfinished cost for each unit than to purchase new machines and pay for their operating costs throughout the time horizon. When the unfinished cost is increased to $100, the opposite effect occurs, and the job shop makes an investment in machines, fewer products are left unfinished. Once the unfinished cost is increased to $100 and above, few products are left unfinished in the integer program’s solution, so similar machine assignments are made. These result in similar numbers of simulated unfinished products remaining at the end of the ramp-up phase, creating a small difference in the total cost.

The areas that have been examined have shown that the production ramp-up system is robust in managing variable inputs when reasonable data are used. For the examination of the system demand, it was determined that the logarithmic ramp-up strategy was the most costly in total, but the least costly at the unit level. On the other hand, the exponential strategy was the least costly in total, but the most expensive at the component level. Purchase and operating costs proved to behave as expected and were not greatly affected in price changes until extreme scenarios were tested. The unfinished cost was the most volatile, and care must be taken to assign a reasonable value that is appropriate for the system. Small changes in this value led to extreme changes in the solution that were correct but may not be desirable in many production environments. Sensitivity analysis here has helped show how the system reacts to changes in the input parameters and determine areas where care must be taken so that it provides an implementable solution.

Chapter 5

Summary, Conclusions and Future Research

5.1 Summary and Conclusions

As the global marketplace expands, an increased numbers of new, innovative products are continuously introduced to the market. It is now more important than ever for manufacturers to gain an advantage over their growing number of competitors in any way possible. Successful companies are characterized by their ability to achieve a high throughput and a low time-to-market on their new product lines. This, however, brings about the challenge of how to optimally transition from low rate initial production to full volume production. The question is not easily answered and inevitably varies from case to case, and the problem becomes even more significant when the total investment on such an endeavor is examined. Companies make huge capital investments in order to produce the supply required by their customers. This poses great risk for the company and gives them the incentive to ramp-up production efficiently. Although, the ramp-up phase will be different depending on industry and type of manufacturing system, exploring the best practices in one system can provide insight into principles that serve a wide range of circumstances.

This thesis investigated the optimal production ramp-up strategy in a job shop manufacturing system while allowing for flexibility by the user. It used simulation and optimization techniques to quickly converge on an optimal, yet robust, solution. This is one of the main advantages of this technique and of pairing optimization with simulation. It yields a solution that has been tested in a stochastic environment and is optimal for the conditions of variability that the simulation presents. Neither optimization nor simulation can work alone to produce such a solution. The system also allows for manipulation by the decision makers, which is critical in a multi-objective optimization problem such as this. This flexibility and the ability to customize the models to a unique job shop system make the system more implementable in a variety of environments.

The case study examined demonstrates the technique's capabilities and applies data to a simulation model presented by Law and Kelton (2000). The case study's results demonstrate the system's ability to quickly converge on a solution and show how decision makers could adjust the solution to meet specific performance goals. Due to the fact that some of the performance goals are inversely related, marginally altering the solution to meet the performance goals of the user can move the solution away from optimal. Though the solution may not be optimal according to the optimization model, it may be preferred by the decision makers according to the significance placed on each performance metric. Lastly, sensitivity analysis was performed in order to test the influence of the data on the results. In general, the data proved to act in a predictable manner.

Through this thesis, a technique has been developed to optimally allocate machines during production ramp-up in a job shop manufacturing system. The technique leads to a solution that is optimal under stochastic conditions, which can be a difficult computation in manufacturing environments that are influenced by numerous sources of variability. This demonstrates the importance of production ramp-up as a growing area of research and why many companies invest in research and development in the field. Companies must take care in producing an effective and economical ramp-up plan that can affect both the total cost of the ramp-up and future earnings.

5.2 Areas for Future Research

Due to the novelty of the problem in modern manufacturing industries, there are many areas in which research on production ramp-up can expand. This thesis has explored ramp-up techniques in a job shop manufacturing system, but there are several other types of manufacturing systems that must ramp-up production as well. Future research could explore techniques to determine optimal machine allocation strategies in these other types of manufacturing systems, including flow shops, continuous assembly lines and flexible manufacturing systems. Studying these types of systems will provide a more comprehensive look at how production ramp-up works as a whole and will enable generalizations to be made.

Other research could examine the changes in ramp-up strategies if one or more of the assumptions made in this study were relaxed. Different trends in machine allocation and ramp-up cost could be found if, for example, labor, material handling, machine failures or spatial constraints were taken into account. These factors would complicate the optimization and simulation models but would make production ramp-up estimates more accurate in real manufacturing environments.

Several performance metrics were taken into account in this study, but some job shops or other manufacturing systems may use other metrics. Some may even want to use one of these metrics as the objective function value and minimize or maximize it in their system instead of total cost. Since multiple metrics are used to determine the performance of the system, multi-criteria optimization techniques could be implemented. These could be implemented directly in the optimization model or simply when the decision makers are deciding between different strategies that they have explored using marginal changes. The optimization techniques would allow the decision makers to prioritize and assign weights to the performance metrics that they feel are most important and would further tailor the results to their individual preferences.

The technique and models used could be implemented into a decision support system, which would make it simpler to use for those who are not experienced in the field of simulation or optimization. Many managers or decisions makers of manufacturing systems may not have a background in these fields, so an easy-to-use package that would allow them to customize the system would help to make the methodology used here more implementable both within and across organizations. Such a package could be invaluable to a company looking to ramp-up its production and would allow it to quickly make key business decisions.

The future research directions in the area of production ramp-up are not limited by those discussed here. These areas only scratch the surface of the possibilities that are available within this expanding research field. This field will become even more important as globalization continues, and competition in global markets continues to grow. Companies will look to quickly develop new products and produce large quantities of them ahead of their competition. These quick changeovers and new product lines will require new manufacturing systems to be built and expanded at a rapid pace. Production ramp-up plans will be needed in order to achieve these production requirements. Companies that adapt and execute these plans effectively will be those that are most successful and can thrive in the new global marketplace.

Appendix A

Optimization Model Code (Lingo 13.0)

```
model:

! define index sets;
sets:
workstation: available, original; !i;
product; !k;
time;!t;

workstationproduct(workstation,product): hours; !ik;
workstationtime(workstation,time): x, purchase_cost, operation_cost; !it;
producttime(product,time): demand, unfinished_cost, u; !kt;
workstationproducttime(workstation,product,time); !ikt;
endsets


! data;
! demand(k,t) = demand for product k in time period t;
! hours(i,k) = hours required in workstation i for each product k;
! purchase_cost(i,t) = cost of purchasing a machine for workstation i in time period t;
! operation_cost(i,t) = cost of operating a machine in workstation i in time period t;
! unfinished_cost(k,t) = cost of not finishing one job of product k in time period t;
! available(i) = hours each machine is available in workstation i;
! original(i) = original number machines in workstation i;
! utilization = target utilization for workstations;


! variables;
! x(i,t) = number of machines in workstation i during period t;
! u(k,t) = number of unfinished products k in time period t;


! (1) objective function = minimize total purchasing, operating and unfinished product
costs;
min = @sum(workstationtime(i,t)| t#eq#1: purchase_cost(i,t)*(x(i,t)-original(i)))
    + @sum(workstationtime(i,t)| t#gt#1: purchase_cost(i,t)*(x(i,t)-x(i,t-1)))
    + @sum(workstationtime(i,t)| t#eq#1: operation_cost(i,t)*x(i,t))
    + @sum(workstationtime(i,t)| t#eq#2: operation_cost(i,t-1)*(x(i,t) - x(i,t-1)) +
      operation_cost(i,t)*(x(i,t-1)))
    + @sum(workstationtime(i,t)| t#eq#3: operation_cost(i,t-2)*(x(i,t) - x(i,t-1)) +
      operation_cost(i,t-1)*(x(i,t-1) - x(i,t-2)) + operation_cost(i,t)*(x(i,t-2)))
    + @sum(workstationtime(i,t)| t#eq#4: operation_cost(i,t-3)*(x(i,t) - x(i,t-1)) +
      operation_cost(i,t-2)*(x(i,t-1) - x(i,t-2)) + operation_cost(i,t-1)*(x(i,t-2) -
      x(i,t-3)) + operation_cost(i,t)*(x(i,t-3)))
    + @sum(workstationtime(i,t)| t#eq#5: operation_cost(i,t-4)*(x(i,t) - x(i,t-1)) +
      operation_cost(i,t-3)*(x(i,t-1) - x(i,t-2)) + operation_cost(i,t-2)*(x(i,t-2) -
      x(i,t-3)) + operation_cost(i,t-1)*(x(i,t-3) - x(i,t-4)) +
      operation_cost(i,t)*(x(i,t-4)))
    + @sum(producttime(k,t): unfinished_cost(k,t)*u(k,t));

! (2) constraint - demand constraint;
@for(workstationtime(i,t)| t#eq#1: utilization*(available(i)*x(i,t)) + @sum(product(k):
hours(i,k)*u(k,t)) > @sum(product(k): hours(i,k)*demand(k,t)));
@for(workstationtime(i,t)| t#gt#1: utilization*(available(i)*x(i,t)) + @sum(product(k):
hours(i,k)*u(k,t)) > @sum(product(k): hours(i,k)*(demand(k,t) + u(k,t-1))));

! (3) constraint - increase number of machines (could add salvage value);
@for(workstationtime(i,t)| t#eq#1: x(i,t) > original(i));
```

```
@for(workstationtime(i,t)| t#gt#1: x(i,t) > x(i,t-1));

! (4) constraint - non-negativity assumed;

! (5) constraint - make x and u integer;
@for(workstationtime(i,t): @gin(x(i,t)));
@for(producttime(k,t): @gin(u(k,t)));


data:

! import data from Excel;

workstation, product, time = @ole('Job Shop Data.xlsm','workstation','product','time');

demand = @ole('Job Shop Data.xlsm','demand');
hours = @ole('Job Shop Data.xlsm','hours');
purchase_cost = @ole('Job Shop Data.xlsm','purchase_cost');
operation_cost = @ole('Job Shop Data.xlsm','operation_cost');
unfinished_cost = @ole('Job Shop Data.xlsm','unfinished_cost');
available = @ole('Job Shop Data.xlsm','available');
original = @ole('Job Shop Data.xlsm','original');
utilization = @ole('Job Shop Data.xlsm','utilization');


! export solution to Excel;

@ole('Job Shop Data.xlsm','output') = x;
@ole('Job Shop Data.xlsm','u') = u;

enddata
end
```

Appendix B

Generated Optimization Model (Lingo 13.0)

```
MODEL:
 [_1] MIN= 500 * X_1_1 + 400 * X_1_2 + 300 * X_1_3 + 200 * X_1_4 + 10100 * X_1_5 +
  500 * X_2_1 + 400 * X_2_2 + 300 * X_2_3 + 200 * X_2_4 + 10100 * X_2_5 + 500 *
  X_3_1 + 400 * X_3_2 + 300 * X_3_3 + 200 * X_3_4 + 10100 * X_3_5 + 500 * X_4_1 +
  400 * X_4_2 + 300 * X_4_3 + 200 * X_4_4 + 10100 * X_4_5 + 500 * X_5_1 + 400 *
  X_5_2 + 300 * X_5_3 + 200 * X_5_4 + 10100 * X_5_5 + 150 * U_1_1 + 150 * U_1_2 +
  150 * U_1_3 + 150 * U_1_4 + 150 * U_1_5 + 150 * U_2_1 + 150 * U_2_2 + 150 * U_2_3
  + 150 * U_2_4 + 150 * U_2_5 + 200 * U_3_1 + 200 * U_3_2 + 200 * U_3_3 + 200 *
  U_3_4 + 200 * U_3_5 - 130000;
 [_2] 192 * X_1_1 + 0.6 * U_1_1 + 0.8 * U_2_1 + 0.7 * U_3_1 >= 1440;
 [_3] 192 * X_2_1 + 0.85 * U_1_1 + 1.2 * U_3_1 >= 990;
 [_4] 192 * X_3_1 + 0.5 * U_1_1 + 0.75 * U_2_1 + U_3_1 >= 1450;
 [_5] 192 * X_4_1 + 1.1 * U_2_1 + 0.9 * U_3_1 >= 1460;
 [_6] 192 * X_5_1 + 0.5 * U_1_1 + 0.25 * U_3_1 >= 400;
 [_7] 192 * X_1_2 - 0.6 * U_1_1 + 0.6 * U_1_2 - 0.8 * U_2_1 + 0.8 * U_2_2 - 0.7 *
  U_3_1 + 0.7 * U_3_2 >= 2880;
 [_8] 192 * X_1_3 - 0.6 * U_1_2 + 0.6 * U_1_3 - 0.8 * U_2_2 + 0.8 * U_2_3 - 0.7 *
  U_3_2 + 0.7 * U_3_3 >= 4320;
 [_9] 192 * X_1_4 - 0.6 * U_1_3 + 0.6 * U_1_4 - 0.8 * U_2_3 + 0.8 * U_2_4 - 0.7 *
  U_3_3 + 0.7 * U_3_4 >= 5760;
 [_10] 192 * X_1_5 - 0.6 * U_1_4 + 0.6 * U_1_5 - 0.8 * U_2_4 + 0.8 * U_2_5 - 0.7 *
  U_3_4 + 0.7 * U_3_5 >= 7200;
 [_11] 192 * X_2_2 - 0.85 * U_1_1 + 0.85 * U_1_2 - 1.2 * U_3_1 + 1.2 * U_3_2 >= 1980;
 [_12] 192 * X_2_3 - 0.85 * U_1_2 + 0.85 * U_1_3 - 1.2 * U_3_2 + 1.2 * U_3_3 >= 2970;
 [_13] 192 * X_2_4 - 0.85 * U_1_3 + 0.85 * U_1_4 - 1.2 * U_3_3 + 1.2 * U_3_4 >= 3960;
 [_14] 192 * X_2_5 - 0.85 * U_1_4 + 0.85 * U_1_5 - 1.2 * U_3_4 + 1.2 * U_3_5 >= 4950;
 [_15] 192 * X_3_2 - 0.5 * U_1_1 + 0.5 * U_1_2 - 0.75 * U_2_1 + 0.75 * U_2_2 - U_3_1
  + U_3_2 >= 2900;
 [_16] 192 * X_3_3 - 0.5 * U_1_2 + 0.5 * U_1_3 - 0.75 * U_2_2 + 0.75 * U_2_3 - U_3_2
  + U_3_3 >= 4350;
 [_17] 192 * X_3_4 - 0.5 * U_1_3 + 0.5 * U_1_4 - 0.75 * U_2_3 + 0.75 * U_2_4 - U_3_3
  + U_3_4 >= 5800;
 [_18] 192 * X_3_5 - 0.5 * U_1_4 + 0.5 * U_1_5 - 0.75 * U_2_4 + 0.75 * U_2_5 - U_3_4
  + U_3_5 >= 7250;
 [_19] 192 * X_4_2 - 1.1 * U_2_1 + 1.1 * U_2_2 - 0.9 * U_3_1 + 0.9 * U_3_2 >= 2920;
 [_20] 192 * X_4_3 - 1.1 * U_2_2 + 1.1 * U_2_3 - 0.9 * U_3_2 + 0.9 * U_3_3 >= 4380;
 [_21] 192 * X_4_4 - 1.1 * U_2_3 + 1.1 * U_2_4 - 0.9 * U_3_3 + 0.9 * U_3_4 >= 5840;
 [_22] 192 * X_4_5 - 1.1 * U_2_4 + 1.1 * U_2_5 - 0.9 * U_3_4 + 0.9 * U_3_5 >= 7300;
 [_23] 192 * X_5_2 - 0.5 * U_1_1 + 0.5 * U_1_2 - 0.25 * U_3_1 + 0.25 * U_3_2 >= 800;
 [_24] 192 * X_5_3 - 0.5 * U_1_2 + 0.5 * U_1_3 - 0.25 * U_3_2 + 0.25 * U_3_3 >= 1200;
 [_25] 192 * X_5_4 - 0.5 * U_1_3 + 0.5 * U_1_4 - 0.25 * U_3_3 + 0.25 * U_3_4 >= 1600;
 [_26] 192 * X_5_5 - 0.5 * U_1_4 + 0.5 * U_1_5 - 0.25 * U_3_4 + 0.25 * U_3_5 >= 2000;
 [_27] X_1_1 >= 3;
 [_28] X_2_1 >= 2;
 [_29] X_3_1 >= 4;
 [_30] X_4_1 >= 3;
 [_31] X_5_1 >= 1;
 [_32] - X_1_1 + X_1_2 >= 0;
 [_33] - X_1_2 + X_1_3 >= 0;
 [_34] - X_1_3 + X_1_4 >= 0;
 [_35] - X_1_4 + X_1_5 >= 0;
 [_36] - X_2_1 + X_2_2 >= 0;
 [_37] - X_2_2 + X_2_3 >= 0;
 [_38] - X_2_3 + X_2_4 >= 0;
 [_39] - X_2_4 + X_2_5 >= 0;
 [_40] - X_3_1 + X_3_2 >= 0;
 [_41] - X_3_2 + X_3_3 >= 0;
```

```
[_42] - X_3_3 + X_3_4 >= 0;
[_43] - X_3_4 + X_3_5 >= 0;
[_44] - X_4_1 + X_4_2 >= 0;
[_45] - X_4_2 + X_4_3 >= 0;
[_46] - X_4_3 + X_4_4 >= 0;
[_47] - X_4_4 + X_4_5 >= 0;
[_48] - X_5_1 + X_5_2 >= 0;
[_49] - X_5_2 + X_5_3 >= 0;
[_50] - X_5_3 + X_5_4 >= 0;
[_51] - X_5_4 + X_5_5 >= 0;
@GIN( X_1_1); @GIN( X_1_2); @GIN( X_1_3); @GIN( X_1_4); @GIN( X_1_5);
@GIN( X_2_1); @GIN( X_2_2); @GIN( X_2_3); @GIN( X_2_4); @GIN( X_2_5);
@GIN( X_3_1); @GIN( X_3_2); @GIN( X_3_3); @GIN( X_3_4); @GIN( X_3_5);
@GIN( X_4_1); @GIN( X_4_2); @GIN( X_4_3); @GIN( X_4_4); @GIN( X_4_5);
@GIN( X_5_1); @GIN( X_5_2); @GIN( X_5_3); @GIN( X_5_4); @GIN( X_5_5);
@GIN( U_1_1); @GIN( U_1_2); @GIN( U_1_3); @GIN( U_1_4); @GIN( U_1_5);
@GIN( U_2_1); @GIN( U_2_2); @GIN( U_2_3); @GIN( U_2_4); @GIN( U_2_5);
@GIN( U_3_1); @GIN( U_3_2); @GIN( U_3_3); @GIN( U_3_4); @GIN( U_3_5);
END
```

Appendix C

Optimization Model Output (Lingo 13.0) – Iteration 1

```
Global optimal solution found.
Objective value:                              1475450.
Objective bound:                              1475450.
Infeasibilities:                              0.000000
Extended solver steps:                             467
Total solver iterations:                          1727


  Export Summary Report
  ---------------------
  Transfer Method:       OLE BASED
  Workbook:              Job Shop Data.xlsm
  Ranges Specified:              1
      output
  Ranges Found:                  1
  Range Size Mismatches:         0
  Values Transferred:           25


  Export Summary Report
  ---------------------
  Transfer Method:       OLE BASED
  Workbook:              Job Shop Data.xlsm
  Ranges Specified:              1
      u
  Ranges Found:                  1
  Range Size Mismatches:         0
  Values Transferred:           15

Model Class:                                      PILP

Total variables:                     40
Nonlinear variables:                  0
Integer variables:                   40

Total constraints:                   51
Nonlinear constraints:                0

Total nonzeros:                     218
Nonlinear nonzeros:                   0

                              Variable           Value        Reduced Cost
                           UTILIZATION       0.8000000            0.000000
                         AVAILABLE( 1)        240.0000            0.000000
                         AVAILABLE( 2)        240.0000            0.000000
                         AVAILABLE( 3)        240.0000            0.000000
                         AVAILABLE( 4)        240.0000            0.000000
                         AVAILABLE( 5)        240.0000            0.000000
                          ORIGINAL( 1)        3.000000            0.000000
                          ORIGINAL( 2)        2.000000            0.000000
                          ORIGINAL( 3)        4.000000            0.000000
                          ORIGINAL( 4)        3.000000            0.000000
```

ORIGINAL(5)	1.000000	0.000000
HOURS(1, 1)	0.6000000	0.000000
HOURS(1, 2)	0.8000000	0.000000
HOURS(1, 3)	0.7000000	0.000000
HOURS(2, 1)	0.8500000	0.000000
HOURS(2, 2)	0.000000	0.000000
HOURS(2, 3)	1.200000	0.000000
HOURS(3, 1)	0.5000000	0.000000
HOURS(3, 2)	0.7500000	0.000000
HOURS(3, 3)	1.000000	0.000000
HOURS(4, 1)	0.000000	0.000000
HOURS(4, 2)	1.100000	0.000000
HOURS(4, 3)	0.9000000	0.000000
HOURS(5, 1)	0.5000000	0.000000
HOURS(5, 2)	0.000000	0.000000
HOURS(5, 3)	0.2500000	0.000000
X(1, 1)	8.000000	500.0000
X(1, 2)	15.00000	400.0000
X(1, 3)	23.00000	300.0000
X(1, 4)	30.00000	200.0000
X(1, 5)	36.00000	10100.00
X(2, 1)	6.000000	500.0000
X(2, 2)	11.00000	400.0000
X(2, 3)	16.00000	300.0000
X(2, 4)	21.00000	200.0000
X(2, 5)	24.00000	10100.00
X(3, 1)	8.000000	500.0000
X(3, 2)	16.00000	400.0000
X(3, 3)	23.00000	300.0000
X(3, 4)	31.00000	200.0000
X(3, 5)	36.00000	10100.00
X(4, 1)	8.000000	500.0000
X(4, 2)	16.00000	400.0000
X(4, 3)	23.00000	300.0000
X(4, 4)	31.00000	200.0000
X(4, 5)	36.00000	10100.00
X(5, 1)	3.000000	500.0000
X(5, 2)	5.000000	400.0000
X(5, 3)	7.000000	300.0000
X(5, 4)	9.000000	200.0000
X(5, 5)	10.00000	10100.00
PURCHASE_COST(1, 1)	10000.00	0.000000
PURCHASE_COST(1, 2)	10000.00	0.000000
PURCHASE_COST(1, 3)	10000.00	0.000000
PURCHASE_COST(1, 4)	10000.00	0.000000
PURCHASE_COST(1, 5)	10000.00	0.000000
PURCHASE_COST(2, 1)	10000.00	0.000000
PURCHASE_COST(2, 2)	10000.00	0.000000
PURCHASE_COST(2, 3)	10000.00	0.000000
PURCHASE_COST(2, 4)	10000.00	0.000000
PURCHASE_COST(2, 5)	10000.00	0.000000
PURCHASE_COST(3, 1)	10000.00	0.000000
PURCHASE_COST(3, 2)	10000.00	0.000000
PURCHASE_COST(3, 3)	10000.00	0.000000
PURCHASE_COST(3, 4)	10000.00	0.000000
PURCHASE_COST(3, 5)	10000.00	0.000000
PURCHASE_COST(4, 1)	10000.00	0.000000
PURCHASE_COST(4, 2)	10000.00	0.000000
PURCHASE_COST(4, 3)	10000.00	0.000000
PURCHASE_COST(4, 4)	10000.00	0.000000
PURCHASE_COST(4, 5)	10000.00	0.000000
PURCHASE_COST(5, 1)	10000.00	0.000000
PURCHASE_COST(5, 2)	10000.00	0.000000
PURCHASE_COST(5, 3)	10000.00	0.000000
PURCHASE_COST(5, 4)	10000.00	0.000000
PURCHASE_COST(5, 5)	10000.00	0.000000
OPERATION_COST(1, 1)	100.0000	0.000000
OPERATION_COST(1, 2)	200.0000	0.000000
OPERATION_COST(1, 3)	300.0000	0.000000

OPERATION_COST(1, 4)	400.0000	0.000000
OPERATION_COST(1, 5)	500.0000	0.000000
OPERATION_COST(2, 1)	100.0000	0.000000
OPERATION_COST(2, 2)	200.0000	0.000000
OPERATION_COST(2, 3)	300.0000	0.000000
OPERATION_COST(2, 4)	400.0000	0.000000
OPERATION_COST(2, 5)	500.0000	0.000000
OPERATION_COST(3, 1)	100.0000	0.000000
OPERATION_COST(3, 2)	200.0000	0.000000
OPERATION_COST(3, 3)	300.0000	0.000000
OPERATION_COST(3, 4)	400.0000	0.000000
OPERATION_COST(3, 5)	500.0000	0.000000
OPERATION_COST(4, 1)	100.0000	0.000000
OPERATION_COST(4, 2)	200.0000	0.000000
OPERATION_COST(4, 3)	300.0000	0.000000
OPERATION_COST(4, 4)	400.0000	0.000000
OPERATION_COST(4, 5)	500.0000	0.000000
OPERATION_COST(5, 1)	100.0000	0.000000
OPERATION_COST(5, 2)	200.0000	0.000000
OPERATION_COST(5, 3)	300.0000	0.000000
OPERATION_COST(5, 4)	400.0000	0.000000
OPERATION_COST(5, 5)	500.0000	0.000000
DEMAND(1, 1)	600.0000	0.000000
DEMAND(1, 2)	1200.000	0.000000
DEMAND(1, 3)	1800.000	0.000000
DEMAND(1, 4)	2400.000	0.000000
DEMAND(1, 5)	3000.000	0.000000
DEMAND(2, 1)	1000.000	0.000000
DEMAND(2, 2)	2000.000	0.000000
DEMAND(2, 3)	3000.000	0.000000
DEMAND(2, 4)	4000.000	0.000000
DEMAND(2, 5)	5000.000	0.000000
DEMAND(3, 1)	400.0000	0.000000
DEMAND(3, 2)	800.0000	0.000000
DEMAND(3, 3)	1200.000	0.000000
DEMAND(3, 4)	1600.000	0.000000
DEMAND(3, 5)	2000.000	0.000000
UNFINISHED_COST(1, 1)	150.0000	0.000000
UNFINISHED_COST(1, 2)	150.0000	0.000000
UNFINISHED_COST(1, 3)	150.0000	0.000000
UNFINISHED_COST(1, 4)	150.0000	0.000000
UNFINISHED_COST(1, 5)	150.0000	0.000000
UNFINISHED_COST(2, 1)	150.0000	0.000000
UNFINISHED_COST(2, 2)	150.0000	0.000000
UNFINISHED_COST(2, 3)	150.0000	0.000000
UNFINISHED_COST(2, 4)	150.0000	0.000000
UNFINISHED_COST(2, 5)	150.0000	0.000000
UNFINISHED_COST(3, 1)	200.0000	0.000000
UNFINISHED_COST(3, 2)	200.0000	0.000000
UNFINISHED_COST(3, 3)	200.0000	0.000000
UNFINISHED_COST(3, 4)	200.0000	0.000000
UNFINISHED_COST(3, 5)	200.0000	0.000000
U(1, 1)	0.000000	150.0000
U(1, 2)	0.000000	150.0000
U(1, 3)	0.000000	150.0000
U(1, 4)	0.000000	150.0000
U(1, 5)	25.00000	150.0000
U(2, 1)	0.000000	150.0000
U(2, 2)	0.000000	150.0000
U(2, 3)	0.000000	150.0000
U(2, 4)	0.000000	150.0000
U(2, 5)	132.0000	150.0000
U(3, 1)	0.000000	200.0000
U(3, 2)	0.000000	200.0000
U(3, 3)	0.000000	200.0000
U(3, 4)	0.000000	200.0000
U(3, 5)	270.0000	200.0000

Row	Slack or Surplus	Dual Price

1	1475450.	-1.000000
2	96.00000	0.000000
3	162.0000	0.000000
4	86.00000	0.000000
5	76.00000	0.000000
6	176.0000	0.000000
7	0.000000	0.000000
8	96.00000	0.000000
9	0.000000	0.000000
10	21.60000	0.000000
11	132.0000	0.000000
12	102.0000	0.000000
13	72.00000	0.000000
14	3.250000	0.000000
15	172.0000	0.000000
16	66.00000	0.000000
17	152.0000	0.000000
18	43.50000	0.000000
19	152.0000	0.000000
20	36.00000	0.000000
21	112.0000	0.000000
22	0.2000000	0.000000
23	160.0000	0.000000
24	144.0000	0.000000
25	128.0000	0.000000
26	0.000000	0.000000
27	5.000000	0.000000
28	4.000000	0.000000
29	4.000000	0.000000
30	5.000000	0.000000
31	2.000000	0.000000
32	7.000000	0.000000
33	8.000000	0.000000
34	7.000000	0.000000
35	6.000000	0.000000
36	5.000000	0.000000
37	5.000000	0.000000
38	5.000000	0.000000
39	3.000000	0.000000
40	8.000000	0.000000
41	7.000000	0.000000
42	8.000000	0.000000
43	5.000000	0.000000
44	8.000000	0.000000
45	7.000000	0.000000
46	8.000000	0.000000
47	5.000000	0.000000
48	2.000000	0.000000
49	2.000000	0.000000
50	2.000000	0.000000
51	1.000000	0.000000

Appendix D

Case Study Marginal Adjustment Iterations

Optimal Solution with System Variability (Iteration 2 as shown in Table 4-15)

Number of Machines

Period	Workstation 1	2	3	4	5
0	3	2	4	3	1
1	8	6	8	8	3
2	15	11	16	16	5
3	23	16	23	23	7
4	30	21	31	31	9
5	38	26	38	38	11

Performance Measures	Result
Unit Cost	$ 51.56
Total Cost	$ 1,546,900.00
Unfinished Products	122.9
% of Total	0.41%

Utilization (%)

Period	Workstation 1	2	3	4	5
0	93.99	97.68	70.62	94.27	78.55
1	75.23	69.82	75.70	76.32	56.41
2	79.75	75.22	75.37	75.96	66.35
3	78.19	76.90	78.65	79.36	70.86
4	79.91	78.18	77.76	78.57	73.59
5	78.80	79.63	79.33	79.74	76.01

Average Number in Queue

Period	Workstation 1	2	3	4	5
0	1.55	10.91	0.14	3.56	0.27
1	0.11	0.48	0.22	0.27	0.08
2	0.14	0.25	0.11	0.10	0.13
3	0.06	0.22	0.14	0.15	0.18
4	0.07	0.21	0.08	0.07	0.22
5	0.03	0.21	0.08	0.07	0.25

Average Time in System (Hours)

Period	Product 1	2	3
0	8.35	4.32	11.19
1	2.74	2.76	4.50
2	2.52	2.68	4.14
3	2.50	2.67	4.12
4	2.49	2.66	4.10
5	2.48	2.66	4.09

Iteration 3 – Remove Machine from Workstation 5

Number of Machines

Period	Workstation 1	2	3	4	5
0	3	2	4	3	1
1	8	6	8	8	3
2	15	11	16	16	5
3	23	16	23	23	7
4	30	21	31	31	9
5	38	26	38	38	10

Performance Measures	Result
Unit Cost	$ 51.23
Total Cost	$ 1,536,800.00
Unfinished Products	122.6
% of Total	0.41%

Utilization (%)

Period	Workstation 1	2	3	4	5
0	93.99	97.68	70.62	94.27	78.55
1	75.23	69.82	75.70	76.32	56.41
2	79.75	75.22	75.37	75.96	66.35
3	78.19	76.90	78.65	79.36	70.86
4	79.91	78.18	77.76	78.57	73.59
5	78.85	79.20	79.31	79.99	83.13

Average Number in Queue

Period	Workstation 1	2	3	4	5
0	1.55	10.91	0.14	3.56	0.27
1	0.11	0.48	0.22	0.27	0.08
2	0.14	0.25	0.11	0.10	0.13
3	0.06	0.22	0.14	0.15	0.18
4	0.07	0.21	0.08	0.07	0.22
5	0.04	0.19	0.09	0.07	0.74

Average Time in System (Hours)

Period	Product 1	2	3
0	8.35	4.32	11.19
1	2.74	2.76	4.50
2	2.52	2.68	4.14
3	2.50	2.67	4.12
4	2.49	2.66	4.10
5	2.51	2.66	4.11

Iteration 4 – Remove Machine from Workstation 1

Number of Machines

Period	Workstation 1	2	3	4	5
0	3	2	4	3	1
1	8	6	8	8	3
2	15	11	16	16	5
3	23	16	23	23	7
4	30	21	31	31	9
5	**37**	26	38	38	10

Performance Measures	Result
Unit Cost	$ 50.88
Total Cost	$ 1,526,500.00
Unfinished Products	122.2
% of Total	0.41%

Utilization (%)

Period	Workstation 1	2	3	4	5
0	93.99	97.68	70.62	94.27	78.55
1	75.23	69.82	75.70	76.32	56.41
2	79.75	75.22	75.37	75.96	66.35
3	78.19	76.90	78.65	79.36	70.86
4	79.91	78.18	77.76	78.57	73.59
5	80.89	79.51	79.35	79.86	83.52

Average Number in Queue

Period	Workstation 1	2	3	4	5
0	1.55	10.91	0.14	3.56	0.27
1	0.11	0.48	0.22	0.27	0.08
2	0.14	0.25	0.11	0.10	0.13
3	0.06	0.22	0.14	0.15	0.18
4	0.07	0.21	0.08	0.07	0.22
5	0.06	0.19	0.09	0.07	0.74

Average Time in System (Hours)

Period	Product 1	2	3
0	8.35	4.32	11.19
1	2.74	2.76	4.50
2	2.52	2.68	4.14
3	2.50	2.67	4.12
4	2.49	2.66	4.10
5	2.51	2.66	4.11

Iteration 5 – Remove Machine from Workstation 3

Number of Machines

Period	Workstation 1	Workstation 2	Workstation 3	Workstation 4	Workstation 5
0	3	2	4	3	1
1	8	6	8	8	3
2	15	11	16	16	5
3	23	16	23	23	7
4	30	21	31	31	9
5	37	26	**37**	38	10

Performance Measures	Result
Unit Cost	$ 50.55
Total Cost	$ 1,516,400.00
Unfinished Products	121.7
% of Total	0.41%

Utilization (%)

Period	Workstation 1	Workstation 2	Workstation 3	Workstation 4	Workstation 5
0	93.99	97.68	70.62	94.27	78.55
1	75.23	69.82	75.70	76.32	56.41
2	79.75	75.22	75.37	75.96	66.35
3	78.19	76.90	78.65	79.36	70.86
4	79.91	78.18	77.76	78.57	73.59
5	80.87	79.30	81.39	79.78	83.41

Average Number in Queue

Period	Workstation 1	Workstation 2	Workstation 3	Workstation 4	Workstation 5
0	1.55	10.91	0.14	3.56	0.27
1	0.11	0.48	0.22	0.27	0.08
2	0.14	0.25	0.11	0.10	0.13
3	0.06	0.22	0.14	0.15	0.18
4	0.07	0.21	0.08	0.07	0.22
5	0.06	0.17	0.14	0.07	0.71

Average Time in System (Hours)

Period	Product 1	Product 2	Product 3
0	8.35	4.32	11.19
1	2.74	2.76	4.50
2	2.52	2.68	4.14
3	2.50	2.67	4.12
4	2.49	2.66	4.10
5	2.50	2.66	4.12

Iteration 6 – Remove Machine from Workstation 2

Number of Machines

Period	Workstation 1	Workstation 2	Workstation 3	Workstation 4	Workstation 5
0	3	2	4	3	1
1	8	6	8	8	3
2	15	11	16	16	5
3	23	16	23	23	7
4	30	21	31	31	9
5	37	**25**	37	38	10

Performance Measures	Result
Unit Cost	$ 50.22
Total Cost	$ 1,506,500.00
Unfinished Products	122.6
% of Total	0.41%

Utilization (%)

Period	Workstation 1	Workstation 2	Workstation 3	Workstation 4	Workstation 5
0	93.99	97.68	70.62	94.27	78.55
1	75.23	69.82	75.70	76.32	56.41
2	79.75	75.22	75.37	75.96	66.35
3	78.19	76.90	78.65	79.36	70.86
4	79.91	78.18	77.76	78.57	73.59
5	80.94	82.41	81.32	79.67	83.60

Average Number in Queue

Period	Workstation 1	Workstation 2	Workstation 3	Workstation 4	Workstation 5
0	1.55	10.91	0.14	3.56	0.27
1	0.11	0.48	0.22	0.27	0.08
2	0.14	0.25	0.11	0.10	0.13
3	0.06	0.22	0.14	0.15	0.18
4	0.07	0.21	0.08	0.07	0.22
5	0.06	0.34	0.14	0.06	0.65

Average Time in System (Hours)

Period	Product 1	Product 2	Product 3
0	8.35	4.32	11.19
1	2.74	2.76	4.50
2	2.52	2.68	4.14
3	2.50	2.67	4.12
4	2.49	2.66	4.10
5	2.51	2.66	4.12

Iteration 7 – Remove Machine from Workstation 4 (Final Solution as shown in Table 4-16)

Number of Machines

Period	Workstation 1	Workstation 2	Workstation 3	Workstation 4	Workstation 5
0	3	2	4	3	1
1	8	6	8	8	3
2	15	11	16	16	5
3	23	16	23	23	7
4	30	21	31	31	9
5	37	25	37	**37**	10

Performance Measures	Result
Unit Cost	$ 49.89
Total Cost	$ 1,496,550.00
Unfinished Products	124.3
% of Total	0.41%

Utilization (%)

Period	Workstation 1	Workstation 2	Workstation 3	Workstation 4	Workstation 5
0	93.99	97.68	70.62	94.27	78.55
1	75.23	69.82	75.70	76.32	56.41
2	79.75	75.22	75.37	75.96	66.35
3	78.19	76.90	78.65	79.36	70.86
4	79.91	78.18	77.76	78.57	73.59
5	80.97	82.27	81.54	82.34	82.79

Average Number in Queue

Period	Workstation 1	Workstation 2	Workstation 3	Workstation 4	Workstation 5
0	1.55	10.91	0.14	3.56	0.27
1	0.11	0.48	0.22	0.27	0.08
2	0.14	0.25	0.11	0.10	0.13
3	0.06	0.22	0.14	0.15	0.18
4	0.07	0.21	0.08	0.07	0.22
5	0.06	0.32	0.14	0.13	0.59

Average Time in System (Hours)

Period	Product 1	Product 2	Product 3
0	8.35	4.32	11.19
1	2.74	2.76	4.50
2	2.52	2.68	4.14
3	2.50	2.67	4.12
4	2.49	2.66	4.10
5	2.51	2.67	4.11

References

Armstrong, Robert J. and Arnoldo C. Hax. “A Hierarchical Approach for a Navel Tender Job-Shop Design,” M.I.T. Operations Research Center, *Technical Report No. 101*, August 1974.

Billington, C. A., and T. C. Davis. "Manufacturing Strategy Analysis: Models and Practice." *Omega* 20.5-6 (1992): 587-95.

Boxma, O., A. H. G. Rinnooy Kan, and M. Van Vliet. "Machine Allocation Problems in Manufacturing Networks." *European Journal of Operational Research* 45.1 (1990): 47-54.

Bowman, E. H. "The Schedule-Sequencing Problem." *Operations Research* 7.5 (1959): 621-24.

Caricato, Pierpaolo, Antonio Grieco, and Francesco Nucci. “Simulation and Mathematical Programming for a Multi-Objective Configuration Problem in a Hybrid Flow Shop.” *Proc. of 2008 Winter Simulation Conference*, Miami, FL.

Carrillo, J. and Gaimon, C., “Improving Manufacturing Performance Through Process Change and Knowledge Creation,” *Management Science* 46.2 (2000): 265-288.

Chan, W. K., and L. W. Schruben. “Properties of discrete event systems from their mathematical programming representations.” *Proc. of 2003 Winter Simulation Conference*. New Orleans, LA.

Cohen, M. A., and S. Moon. "Impact of Production Scale Economies, Manufacturing Complexity and Transportation Costs on Supply Chain Facility Networks." *Journal of Manufacturing and Operations Management* 3 (1990): 269-92.

Couretas, J. M., B. P. Zeigler, and G. V. Mignon. "SEAE-SES Enterprise Alternative Evaluator: Design and Implementation of a Manufacturing Enterprise Alternative Evaluation Tool."

Proceedings of SPIE: *Enabling Technology for Simulation Science III*. Vol. 3696. 1998. 136-46.

Couretas, J. M., B. P. Zeigler, and G. V. Mignon. "SEAE-SES Enterprise Alternative Evaluator: a Comprehensive Manufacturing Process Design Framework." *15th ISPE/IEEE International Conference on CAD/CAM, Robotics and Factories of the Future: CARS & FOF '99*. Aguas De Lindoia, Brazil. Los Alamitos, CA: IEEE, 1999.

Couretas, J. M., B. P. Zeigler, I. Subramanian, and H. Sarjoughian. "Capacity Analysis for Mixed Technology Production: Evaluating Production Ramp Resource Modifications Via Distributed Simulation." *International Journal of Production Research* 39.2 (2001): 163-84.

Cox, James F., and Fred P. Adams. "Manufacturing Resource Planning: an Integrated Decision Support System." *Simulation* 35.3 (1980): 73-79.

Dallery, Yves, and Kathryn E. Stecke. "On the Optimal Allocation of Servers and Workloads in Closed Queueing Networks." *Operations Research* 38.4 (1990): 694-703.

Dallery, Yves, and Yannick Frein. "An Efficient Method to Determine the Optimal Configuration of a Flexible Manufacturing System." *Annals of Operations Research* 15.1 (1988): 207-25.

de Boer, R., J.M.J. Schutten, and W.H.M. Zijm. "A Decision Support System for Ship Maintenance Capacity Planning." *CIRP Annals - Manufacturing Technology* 46.1 (1997): 391-96.

Du, Shichang, Lifeng Xi, Jun Ni, Pan Ershun, and C. Richard Liu. "Product Lifecycle-Oriented Quality and Productivity Improvement Based on Stream of Variation Methodology." *Computers in Industry* 59.2-3 (2008): 180-92.

Dyer, M. E., and L. G. Proll. "On the Validity of Marginal Analysis for Allocating Servers in M/M/c Queues." *Management Science* 23.9 (1977): 1019-022.

Eklin, Mark, Yohanan Arzi, and Avraham Shtub. "Model for Cost Estimation in a Finite-capacity Stochastic Environment Based on Shop Floor Optimization Combined with Simulation." *European Journal of Operational Research* 194.1 (2009): 294-306.

Feldman, P., and A. Shtub. "Model for Cost Estimation in a Finite-capacity Environment." *International Journal of Production Research* 44.2 (2006): 305-27.

Fox, B. "Discrete Optimization via Marginal Analysis." *Management Science* 13.3 (1966): 210-16.

Frenk, Hans, Martine Labbe, Mario Van Vliet, and Shuzhong Zhang. "Improved Algorithms for Machine Allocation in Manufacturing Systems." *Operations Research* 42.3 (1994): 523-30.

Fu, Michael C. "Are We There Yet? the Marriage between Simulation & Optimization." *OR/MS Today* (2007): 16-17.

Fu, Michael C., Chun-Hung Chen, and Leyuan Shi. “Some Topics for Simulation Optimization.” *Proc. Of 2008 Winter Simulation Conference*, Miami, FL.

Greenwood, Allen G., Sucharith Vanguri, Burak Eksioglu, Pramod Jain, Travis W. Hill, Jeffery W. Miller, and Clayton T. Walden. "Simulation Optimization Decision Support System for Ship Panel Shop Operations." *Proc. of 2005 Winter Simulation Conference*, Orlando, FL.

Haller, Martin, Andreas Peikert, and Josef Thoma. "Cycle Time Management during Production Ramp-up." *Robotics and Computer-Integrated Manufacturing* 19 (2003): 183-88.

Harmonosky, Catherine M., Jennifer L. Miller, Scott L. Rosen, Mark T. Traband, Rick Tillotson, and Dave Robbie. "Interfacing Simulation with Costing Software to Drive the Transformation from Prototype Manufacturing to High Volume Manufacturing."*Proc. of 1999 Winter Simulation Conference*, Phoenix, AZ.

Hopp, Wallace J., and Mark L. Spearman. *Factory Physics*. 3rd ed. New York, NY: McGraw-Hill, 2008.

Huang, Ming-Guan, Pao-Long Chang, and Ying-Chyi Chou. "Forward Recursive Scheme with Improved Marginal Analysis Heuristic for Machine Expansion Scheduling in a New Job Shop." *Computers & Industrial Engineering* 50.1-2 (2006): 148-60.

Keen, Peter G. W. "Adaptive Design for Decision Support Systems." *ACM SIGOA Newsletter* 1.4-5 (1980a): 15-25.

Keen, Peter G. W. *Decision Support Systems and Managerial Productivity Analysis*. Cambridge, Mass.: Center for Information Systems Research, Sloan School of Management, MIT, 1980b.

Klemmt, Andreas, Sven Horn, Gerald Weigert. "Investigation of Modified Heuristic Algorithms for Simulation-Based Optimization." *Papers of the 30th International Spring Seminar on Electronics Technology*. 2007. 24-29.

Klemmt, Andreas, Sven Horn, Gerald Weigert, and Klaus-Jürgen Wolter. "Simulation-based Optimization vs. Mathematical Programming: A Hybrid Approach for Optimizing Scheduling Problems." *Robotics and Computer-Integrated Manufacturing* 25.6 (2009): 917-25.

Koren, Y., U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Vanbrussel. "Reconfigurable Manufacturing Systems." *CIRP Annals - Manufacturing Technology* 48.2 (1999): 527-40.

Law, Averill M., and W. David Kelton. *Simulation Modeling and Analysis*. 3rd ed. Boston: McGraw-Hill, 2000.

Manne, Alan S. "Capacity Expansion and Probabilistic Growth." *Econometrica* 29.4 (1961): 632-49.

Manne, Alan S. *Investments for Capacity Expansion; Size, Location, and Time-phasing,*. Cambridge: M.I.T., 1967.

Manne, Alan. S. "On the Job-Shop Scheduling Problem." *Operations Research* 8.2 (1960): 219-23.

Matta, Andrea, Maurizio Tomasella, and Anna Valente. "Impact of Ramp-Up on the Optimal Capacity-Related Reconfiguration Policy." *International Journal of Flexible Manufacturing Systems* 19.3 (2007): 173-94.

Matta, Andrea. “Simulation Optimization with Mathematical Programming Representation of Discrete Event Systems.” *Proc. of 2008 Winter Simulation Conference*, Miami, FL.

Pinedo, Michael. *Planning and Scheduling in Manufacturing and Services*. New York, NY: Springer, 2009.

Power, D. J. *Decision Support Systems Resources*. Web. 15 Feb. 2011. <http://dssresources.com>.

Power, D., and R. Sharda. "Model-driven Decision Support Systems: Concepts and Research Directions." *Decision Support Systems* 43.3 (2007): 1044-061.

Rolfe, A. J. “A note on marginal allocation in multiple server service systems.” *Management Science* 17.9 (1971): 656–658.

Rosen, Scott L., Catherine M. Harmonosky, and Mark T. Traband. "Optimization of Systems with Multiple Performance Measures via Simulation: Survey and Recommendations." *Computers and Industrial Engineering* 54.2 (2008): 327-39.

Rubinstein, Reuven Y. "Sensitivity Analysis and Performance Extrapolation for Computer Simulation Models." *Operations Research* 37.1 (1989): 72-81.

Schriber, Thomas J. and Kathryn E. Stecke. “Using Mathematical Programming and Simulation to Study FMS Machine Utilizations.” *Proc. of 1987 Winter Simulation Conference*, Atlanta, GA.

Schruben, L. W. “Mathematical Programming Models of Discrete Event System Dynamics.” *Proc. of 2000 Winter Simulation Conference*. Orlando, FL.

Shim, J. P., Merrill Warkentin, James F. Courtney, Daniel J. Power, Ramesh Sharda, and Christer Carlsson. "Past, Present, and Future of Decision Support Technology." *Decision Support Systems* 33.2 (2002): 111-26.

“Simio Reference Guide” 3rd ed. Simio LLC (2010).

Takakuwa, Soemon. "The Use of Simulation in Activity-Based Costing for Flexible Manufacturing Systems." *Proc. of 1997 Winter Simulation Conference*. Atlanta, GA.

Terwiesch, Christian, and Roger E. Bohn. “Learning and Process Improvement During Production Ramp-Up.” *International Journal of Production Economics* 70.1 (2001): 1-19.

Terwiesch, Christian, and Yi Xu. "The Copy-Exactly Ramp-Up Strategy: Trading-Off Learning With Process Change." *IEEE Transactions on Engineering Management* 51.1 (2004): 70-84.

Van Vliet, Mario, and Alexander H. G. Rinnooy Kan. "Machine Allocation Algorithms for Job Shop Manufacturing." *Journal of Intelligent Manufacturing* 2.2 (1991): 83-94.

Wagner, Harvey M. "An Integer Linear-programming Model for Machine Scheduling." *Naval Research Logistics Quarterly* 6.2 (1959): 131-40.

Weber, R. W. “On the marginal benefit of adding server to GI/G/m queues.” *Management Science*, 26.9 (1980): 946–951.

Yan, Yan, and Guoxin Wang. "A Job Shop Scheduling Approach Based on Simulation Optimization." *Proc. of IEEM 2007: The IEEE International Conference on Industrial Engineering and Engineering Management*, 2-5 December 2007, Singapore.

Zeigler, Bernard P. *Multifaceted Modeling and Discrete Event Simulation*. London: Academic Press, 1984.

Zeigler, Bernard P. *Object-oriented Simulation with Hierarchical, Modular Models: Intelligent Agents and Endomorphic Systems*. Boston: Academic Press, 1990.

Zeigler, Bernard P., and Herbert Praehofer. *Theory of Modeling and Simulation*. 2nd ed. New York: Academic Press, 2000.