TWO PAPERS IN SUPPLY CHAIN DESIGN: SUPPLY CHAIN CONFIGURATION AND PART SELECTION IN MULTIGENERATION PRODUCTS

by

Sean Peter Willems

B.S.E. Economics
Wharton School, University of Pennsylvania, 1993

S.M. Operations Research
Massachusetts Institute of Technology, 1996

Submitted to the

Alfred P. Sloan School of Management

in partial fulfillment of the requirement for the degree of

Doctor of Philosophy in Management

at the
Massachusetts Institute of Technology
February 1999

Signature of Author ____
Alfred P. Sloan School of Management
January 26, 1999

Certified by ____
Stephen C. Graves
Abraham J. Siegel Professor of Management
Co-director, Leaders for Manufacturing Program

Accepted by ____

This page intentionally left blank

TWO PAPERS IN SUPPLY CHAIN DESIGN: SUPPLY CHAIN CONFIGURATION AND PART SELECTION IN MULTIGENERATION PRODUCTS

by

Sean Peter Willems

Submitted to the Alfred P. Sloan School of Management
on January 26, 1999 in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy in Management

Abstract

Increasing competitive pressures are forcing companies to increase their rates of innovation. The increasing rate of innovation shortens each product's duration in the market, thereby compressing each product's life cycle. Without proper management, increasing product turnover will increase design and manufacturing costs. More frequent product development cycles require additional product development resources. Shorter production runs inhibit a company's ability to achieve manufacturing cost reductions by exploiting the learning curve and scale economies. Unless companies can efficiently manage multiple generations of the product, there is a substantial risk that costs will spiral out of control.

Focusing on supply chain design is one way companies can combat the problems caused by increased competition and shorter product life cycles. Supply chain design attempts to create the appropriate supply chain for the company's operating environment. This dissertation addresses two problems that are relevant to supply chain design.

The first problem addresses how to configure a new product's supply chain. In this problem, the product's design has already been fixed. The central question is determining what parts and processes to select. For example, various vendors can supply a certain raw material, multiple machines or processes can manufacture the assembly, and multiple shipping options can deliver the completed product to the final customer. Each of these different options is differentiated by its production time and direct cost added. Given these various choices along the supply chain, the problem is to select the options that minimize the total supply chain cost.

The second problem addresses part selection for multigeneration products. When product life cycles are short, the company could opt to overengineer certain components or subsystems in the current generation. This would increase the current period's costs but allow the company to forgo a development cycle in the next period and gain cost efficiencies by exploiting the higher volume from two generations of demand. This research considers development costs, manufacturing costs and part functionality requirements in order to determine the optimal upgrade path for components across multiple product generations.

Thesis Supervisor: Stephen C. Graves
Title: Professor of Management Science

This page intentionally left blank

Acknowledgments

I have worked with Steve Graves for five years now. I will be forever grateful to Steve for giving me the opportunity to achieve my potential at MIT. I can't even begin to articulate how much I have learned from him.

Steve Eppinger and Yashan Wang formed the rest of my committee. They were both very giving of their time and energy in order to improve this research. I sincerely appreciate the direction and feedback that they provided.

I gratefully acknowledge the support and resources made available to me through the Leaders for Manufacturing Program, a partnership between MIT and major U.S. manufacturing companies. It would be impossible to overstate how much of an impact LFM has had on my research. The sponsor companies that I have interacted with have been very willing to spend the time and effort required to help make my research relevant to their problems.

The close bond I formed with several classmates significantly enhanced my learning at MIT. David Kletter, Steve's senior student when I first came to MIT, was the model student that I wanted to become. He took the time to make sure that I really understood the concepts that I thought I understood. John Ruark, probably the smartest non-faculty member I have ever met, taught me a tremendous amount about operations research and how to make it accessible. Brian Tomlin was the close friend that I progressed through the program with.

This dissertation represents a tremendous achievement for me and my wife Anjanette. Although it sounds trite, it is so true that without her support I could not have done this.

This page intentionally left blank

To my father

This page intentionally left blank

Table of Contents

Part II - Part Selection in Multigeneration Products

1 Introduction

1.1 Problem context

Increasing competitive pressures are forcing companies to increase their rates of innovation. The increasing rate of innovation shortens each product's duration in the market, thereby compressing each product's life cycle. Product categories, like networking equipment, which used to have a life span of two to four years are now obsolete after one to two years.

Without proper management, increasing product turnover will increase design and manufacturing costs. More frequent product development cycles require additional product development resources. Shorter production runs inhibit a company's ability to achieve manufacturing cost reductions by exploiting the learning curve and scale economies. Unless companies can efficiently manage multiple generations of the product, there is a substantial risk that costs will spiral out of control.

This dissertation addresses two problems that are relevant to supply chain design. The first problem is how to configure a new product's supply chain. In this problem, there are various vendors that can supply a certain raw material, multiple machines or processes that can manufacture the base subassembly, and multiple shipping options to deliver the completed product to the final customer. Each of these different options is differentiated by its production time and direct cost added. For example, the company has two available options to deliver the product to the final customer. The product can be sent by air at a unit cost of $20 and a transportation time of two days or by truck at a unit cost of $8 with a four-day transit time. Given these various choices along the supply chain, the problem is to select the options that minimize the total supply chain cost.

The second problem addresses part selection for multigeneration products. When product life cycles are short, the company could opt to overengineer certain components or subsystems in the current generation. This would increase the current period's costs but allow the company to forgo a development cycle in the next period and gain cost efficiencies by exploiting the higher volume from two generations of demand.

These are by no means the only relevant problems when designing supply chains. Another significant problem considers the impact of quality on the supplier-manufacturer relationship. Examples of this work include work on optimal sampling policies for incoming parts (Nurani et al. (1995)), illustrating the benefits of highly robust part designs (Clausing (1993)), and calculating the cost of quality implications from poor quality vendors. Another stream of research has worked on ensuring that the supplier is capable of meeting the manufacturer's volume requirements. This research, which falls broadly under the category of supplier certification (Grieco and Gozzo (1992)), designs practices to ensure that the supplier is able to keep up as the manufacturer ramps up production. There are also relevant strategic issues including the proper organizational structure for complex supply chains (Laseter (1998)).

An essential element of both of the problems presented in this dissertation is that they develop quantitative models that advance the state of decision making beyond the consideration of just unit manufacturing cost (UMC). UMC is defined here as the per unit cost of a completed finished goods item. This is typically the sum of two sets of costs: direct costs and allocated overhead. As the name implies, direct costs are those costs that can be directly attributed to the production of the product. Examples include raw material, transportation, and processing costs. Overhead costs include those costs that are necessary to support the product family but can not be attributed to specific units of production. Examples of engineering costs include quality auditing and process engineering costs. These two components of UMC are often treated independently. That is, engineering costs are based on volume projections for the product and then divided by the total number of units sold in order to yield a per unit cost. Direct costs are just the sum of the direct costs added across the supply chain. In this dissertation, UMC will refer to just the direct portion of the product's cost.

In practice, UMC is the dominant criterion in the design of a supply chain for several reasons. First, in most structured product development processes a product must achieve a gross margin target before it gains approval. Gross margin is calculated by dividing the difference between selling price and UMC by the selling price. Since the price is typically dictated by exogenous factors, the team must focus on UMC in order to meet the gross margin target. This gives UMC a disproportionate influence during the product development process. Second, it can be measured directly, without any ambiguity. This is in contrast to the calculation of costs like expected safety stock cost which require an estimate of quantities like demand variability. Since, by construction, these estimates of variability are only estimates, the actual realization of the safety stock cost will almost surely deviate from its expected value. Although materials managers understand this fact, it nonetheless greatly complicates their budgeting process if they choose to include safety stock cost since it will make their budget numbers incorrect. UMC does not have this problem because it equals the sum of costs that are specified in contracts; its value will only change if a change in a contract is negotiated. Third, the buyers that negotiate the part purchases are not the same employees that deal with the consequences of the purchasing decisions. They do not see the effects of choosing a cheaper, less responsive, supplier. Finally, few quantitative tools exist to assess the impact of UMC-based decisions on the rest of the supply chain. In the absence of a model that directly proves that minimizing UMC is a bad decision rule, they will continue applying a rule that they know with certainty minimizes one cost: the cost-of-goods sold.

The rest of this chapter discusses each problem in more detail. The objective is to frame the problem in light of previous research, discuss the solution technique, and articulate why the problem is worth studying.

1.2 Optimal supply chain configuration strategies for new products

A supply chain can be viewed as a network where the nodes represent functionality that must be provided and the arcs capture precedence constraints among the functions. A function might be the procurement of a raw material, the manufacture of an assembly, or the shipment of a product to a distribution center. For each of these functions, there are one or more options available to satisfy the function. As an example, two options might exist for the procurement of a resistor: a high cost local distributor and a lower cost multinational manufacturer.

For most structured product development processes, there comes a time when the materials management organization (MMO) is called in to source the new product's supply chain. The supply chain's functions have already been determined at this point. The role of MMO is to identify the options that can satisfy each function and then to decide which options to select. A question MMO faces is whether to create a higher unit manufacturing cost, but more responsive, supply chain versus a lower manufacturing cost, less responsive supply chain.

We define a configuration as an assignment where each function is assigned an option. If there are N functions and two options per function, then there are 2^N possible supply chain configurations. This research focuses on finding the optimal supply chain configuration.

Since the supply chain has not yet been established, numerous costs are configuration-dependent. That is, the options selected will impact multiple supply chain design costs. The most obvious cost is the cost-of-goods sold where COGS is defined as the direct variable cost of the product time the number of units sold. Traditionally, sourcing decisions have minimized this cost by minimizing the UMC of the product. In most industrial contexts, minimizing UMC is equivalent to minimizing COGS because UMC can be broken into two components: material-related costs and engineering-related costs. If both types of costs are independent then COGS equals the direction portion of UMC times the number of units sold. However, there are also other costs that are determined by the configuration of the supply chain. These costs include the cost of the inventory necessary to provide the desired level of service to the customer. In particular, this model quantifies both pipeline and safety stock inventory holding costs, based on the costs and production times for the configuration, as well as the customer demand requirements.

This model seeks to optimize jointly the supply chain's cost-of-goods-sold and inventory holding costs. To the best of my knowledge, these costs have not previously been considered jointly. Certain aspects of the problem have been extensively studied in the areas of multi-echelon inventory theory and product development, but these studies have only looked at part of the problem being addressed in this work.

On the multi-echelon inventory side, numerous papers address optimizing safety stock placement across the supply chain. The papers that are the most relevant to our work are the papers by Ettl et al. (1996) and Graves and Willems (1998). These papers, and many of their cited references, optimize safety stock levels for an established supply chain. That is, these models consider supply chains that are already in existence. Because these supply chains are established, several of the relevant costs in the supply chain are already fixed. In particular, the expected pipeline stock cost and COGS are determined by the problem's inputs. Therefore, these costs are constants that do not enter into the analysis.

Inderfurth (1993) does jointly consider the optimization of safety stock costs and production times for a supply chain where the final production stage produces multiple end items. The optimization captures the impact that the finished goods' lead-times have on the overall safety stock cost in the supply chain. However, the model only considers changing the configuration at one stage in the supply chain and only considers safety stock costs.

Assessing the time-to-market costs for new products has also been well studied. Quantitative models that evaluate the cost of development times are considered in Ulrich et al. (1993) and Cohen et al. (1996). Several empirical studies also quantify the benefits of a shorter time-to-market. Hendricks and Singhal (1997) have examined the impact of delayed product introductions on firms' stock price and Datar et al. (1997) have shown that faster product development correlates positively with increased market share in the computer component industry.

To summarize, this model seeks to minimize safety stock cost, pipeline stock cost, and cost of goods sold when sourcing a new supply chain. To accomplish this, a two-state dynamic program is formulated. This formulation is an extension of the model in Graves and

Willems (1998). In Graves and Willems (1998), a single-state dynamic program is formulated to minimize the safety stock cost in an existing supply chain. The single-state formulation allows each stage in the supply chain to be solved with only local information from adjacent stages; this formulation significantly reduces the computational complexity of the problem. This dissertation extends the approach to the case where there are now multiple options at each stage in the supply chain. Algorithms are presented to solve assembly, distribution, and spanning trees. An example of this model's application at Kodak is provided in order to illustrate the trade-offs that this model captures.

1.3 Optimal part selection strategies in multigeneration products

A product can be thought of as consisting of two sets of parts: custom parts and standard parts. In networking equipment, custom parts include microprocessors and ASICs. These are the parts that dictate the performance level of the product and hence the product's relative attractiveness to consumers. Standard parts include memory and system boards. These parts act to support the custom parts; they are necessary for the operation of the product, but they do not differentiate the product from the competition. Because custom parts dictate the product's performance, typically they will have to be revised each generation. However, there is no such requirement on standard parts. Standard parts only need to be revised when they conflict with custom parts or constrain the performance of the product or when they become too expensive. For example, 66 megahertz system boards become impractical to use when microprocessor speeds exceed 333 megahertz since the board's slow speed acts to constrain the entire product's performance thereby negating the effect of the improved microprocessor. With regards to cost, many standard parts are commodities and as such can become prohibitively expensive to use as supply options decrease over time; this explains the move from four-megabyte memory chips to sixteen-megabyte memory chips.

Another fact that complicates this problem is the fact that performance requirements for future periods are uncertain. For example, a computer manufacturer may know that the current product requires a 266 megahertz processor but there may be significant uncertainty whether the requirement for the next period will exceed 333 megahertz. In the unlikely event that the design team knows that the next generation processor will not exceed 333 megahertz, then the 66 megahertz system board will be their choice for the current generation. But if there is a high likelihood that the next generation's processor will exceed 333 megahertz, it may be cost-effective to switch to the 100 megahertz system board in the current generation.

The exact part that will be chosen each period will depend on the cost of changing the part from one period to the next, the per generation part cost, and the level of uncertainty with regards to future performance requirements. To solve this problem, a multi-period two-state backward dynamic programming problem is formulated and solved.

This research problem has been expressed in terms of part selection due to the industrial context that is motivating the problem. As far as I know there has been no work that specifically addresses the problem that I have posed. However, the product development, equipment selection, and technology choice domains have all addressed aspects of this problem.

In the product development domain, Sanderson (1991) presents a stylized model that examines the cost implication of different ratios of standard to custom parts. The cost function is composed of product development, fixed equipment and variable manufacturing costs. Both the development and manufacturing costs depend on the ratio of standard to custom parts. For a given ratio of standard to custom parts, the cost function can be examined in order to identify ranges where adopting one set of parts versus another is beneficial. In contrast, our model's focus is at a lower level of detail. Given a function's requirements over several generations, we are concerned with finding the optimal set of parts to meet the requirements at a minimum cost. For a single product generation, Krishnan et al. (1998) determines the optimal set of common parts and the optimal number of products to offer that maximizes the product family's profitability. The attractiveness of a product offering, and hence its profitability, is dictated by the product's performance level. The authors show that the problem can be converted into a shortest path problem. Because Krishnan et al. (1998) is a single generation problem, the requirements for the period are known with certainty and as such only appear in the demand function. In contrast, our model focuses on the multi-generation issues when making a sourcing decision in the current period.

In the equipment replacement problem, a decision has to be made to either replace a machine in the current period or to replace it in a future period. The classic work in this field is by Terborgh (1949). Relevant factors affecting this decision are the cost of operating the existing machine and the operating characteristics of machines that will become available in the future. For the deterministic version of the problem, extensive planning horizon results have been developed, leading to the creation of efficient forward dynamic programs; refer to Chand and Sethi (1982) for an example of this approach.

The part selection problem differs in several ways from the equipment replacement problem. First, equipment replacement focuses on the role of depreciation and salvage costs for the existing piece of equipment. There is no real analog to these costs in the part selection problem. Second, the equipment replacement models assume that any piece of equipment is suitable to the task at hand, albeit with different operating costs. This may be a reasonable assumption for machinery but it is not a reasonable assumption for parts. That is, there comes a time when some parts are simply unable to satisfy a generation's performance requirement. Finally, equipment replacement models assume that the cost of different machines move in lock step. That is, a newer machine is cheaper to operate than an older machine. A corollary to this is that the newer machine will not become more expensive to operate than older machines when the newer machine is no longer the newest; the assumption here is that costs tend to move together. For the part selection problem, this is not the case. In fact, different technologies can have radically different cost profiles over time. Understanding the impact of these cost differences is a key focus of this research.

The technology choice literature is distinct from the equipment replacement literature in its focus on determining which technology to use given the characterization of demand and the fixed and variable costs associated with each technology. In the equipment replacement problem, the objective is to minimize cost whereas in the technology choice literature the objective is to maximize revenue. The two approaches are not equivalent because the technology literature allows the choice of technology to impact the firm's profitability. Examples of this work include Cohen and Halperin (1986) and Fine and Freund (1990).

In Cohen and Halperin (1986), the authors develop a model that jointly optimizes production levels and the technology chosen each period. That is, different technologies will have different optimal production levels. The authors provide a condition that, if satisfied, proves that any future technology adopted will have a lower per unit variable cost than the existing technology in place. The part costs in our model violate the condition in Cohen and Halperin (1986).

Oftentimes, technology choice considers flexible manufacturing equipment that is capable of satisfying demand from multiple end items. Fine and Freund (1990) consider a multiproduct firm that has a two-stage decision process. In the first stage, they must determine which technologies to adopt. They can choose flexible equipment, at a higher cost, that can produce any product or they can choose dedicated equipment that can only produce a certain product. In the second stage, the capacity purchased in the first period is used to satisfy demand. The authors solve this as a nonlinear stochastic program and prove properties of the optimal solution.

Finally, the work of Rajagopalan et al. (1998) deserves special consideration due to its close relation to our work. The authors present a model that jointly optimizes technology choice, capacity, and replacement decisions. The sources of uncertainty in their model are the times between technology breakthroughs and the new technology that will emerge. This is modeled as a semi-markov process where the transition from one technology to the next is dependent on the best technology currently available. Assumptions are that the set of possible technologies is known at the start of the planning horizon, that the technologies can be indexed from worst to best, and that the likelihood of moving from one technology to each superior technology can be assigned a probability by the user. That is, at the start of the horizon the user needs to specify the possible interrelationships between all technologies. Our model differs in that it focuses on what level of functionality the part must provide, not on which technology will provide the functionality. For example, when a supporting part like a system board is being designed, it is only relevant to know that Moore's law still holds. It is not necessary to know which specific technology in the future leads to this realization of Moore's law. Furthermore, for several of the examples we are using, the constraint in the future is not technological, but a requirement by the consumer. In the example we will see later in this dissertation, the performance constraint is the size of the product. Each generation, the same plastics will be used. The key issue is whether it is optimal to purchase a smaller, but more expensive, circuit board.

Thus, the part selection problem can be viewed as a more constrained version of the technology choice problem. The added constraint is the per period performance requirement. An example where this type of constraint would be useful to the equipment replacement or technology choice literature is for a product like a photolithography machine. In the current generation, machines capable of .25 micron widths are sufficient but in two periods they will need to be capable of .18 micron widths. The question is do you buy a .25 micron width machine now and install a .18 micron width machine in two periods or do you install the .18 micron now; it is assumed that the .18 machine is capable of producing .25 micron widths. If you install the .18 micron machine now, your current period costs will increase but they may be offset by the scale economies achieved by the .18 machine.

Part I : Optimal Supply Chain Configuration Strategies for New Products

2 Introduction

This project will look at determining optimal configuration strategies for new product supply chains. The goal is to develop a decision support tool that product managers can use during the product development process where the product's design has been fixed, but the vendors, manufacturing technologies, and shipment options have not yet been determined. Our supply chain design framework considers three specific costs that are relevant when designing new supply chains: unit manufacturing cost, safety stock cost, and pipeline stock cost. The supply chain design problem minimizes the sum of these costs when creating a new supply chain.

The problem is a design problem because there are several available sourcing options at each stage. Examples include multiple vendors available to supply a raw material and several manufacturing processes capable of assembling the finished product. These different options have different direct costs and production lead-times. Therefore, choices in one portion of the supply chain can affect the costs and responsiveness of the rest of the supply chain. The optimal configuration of the supply chain will choose one option per stage such that the costs of the resulting supply chain are minimized.

The paper is structured as follows. In Section 3, the algorithm for a serial line supply chain is formulated. This section forms the building block of the more general algorithm. Section 4 extends the serial line framework to solve assembly networks. Assembly networks are networks where a stage can have several suppliers but can itself supply only one stage. Section 5 extends the serial line framework to solve distribution networks. Distribution networks are networks where a stage can have multiple customers but only one supplier.

Section 6 combines the results from Sections 4 and 5 to solve spanning tree networks. While still having a specialized structure, spanning trees allow the modeling of numerous real-world supply chains. In Section 7, the algorithm and several heuristics are applied to an industry example.

3 Serial Line Formulation

3.1 Network Representation

We consider an N-stage serial system, where stage i is the immediate upstream supplier for stage i+1, for i = 1, 2, …, N-1. Hence, stage 1 is the raw material stage and has no supplier; and stage N is the finished goods inventory stage, from which customer demand is served. Each stage represents a major processing function in the supply chain. A typical stage might represent the procurement of a raw material or the manufacturing of a subassembly or the shipment of the finished product from a regional warehouse to the customer's distribution center. Each stage is a potential location for holding a safety-stock inventory of the item processed at the stage.

For each stage, one or more options exist that can satisfy the stage's processing requirement. We denote the total number of options available at stage i by O_i. For example, if a stage represents the procurement of a metal housing, then one option might be a locally-based high-cost provider and another option could be a low-cost multinational company. If these are the only two options available at stage 1, then $O_1 = 2$ and the individual options will be denoted O_{11} and O_{12} where O_{ij} denotes the jth available option at stage i. Options are differentiated by their direct costs and production lead-times. For each stage, only one option will be used in the completed supply chain. Thus, this model restricts itself to sole sourcing at each stage.

A schematic of a typical supply chain is shown in Figure 3-1:

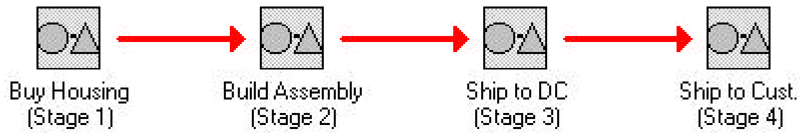

Figure 3-1: Depiction of a typical serial supply chain

and the options at each stage are shown in Table 3-1:

Stage	Option	Description	Direct Cost	Lead-time
1	1	Local supplier	$45	20 days
1	2	Multinational supplier	$20	40 days
2	1	Manual assembly	$10	10 days
2	2	Automated assembly	$40	2 days
2	3	Hybrid assembly line	$20	4 days
3	1	Company-owned trucks	$15	4 days
3	2	Third party carrier	$30	2 days
4	1	Ground transportation	$25	5 days
4	2	Air freight	$45	3 days
4	3	Premium air freight	$60	1 day

Table 3-1: Options available for supply chain example

Figure 3-1 represents a stylized supply chain where a raw material is purchased from an external vendor, transformed and then sent through the company's distribution center. Customers then place orders directly on the distribution center to receive the product. Table 3-1 lists the options available at each stage in the supply chain.

Just looking at the numbers, it is not immediately obvious which option should be selected at each stage. At the extremes, one can create a high-cost, short production lead-time supply chain or a low-cost, long production lead-time supply chain. The goal of this research is twofold. First, for a given set of options, we want to provide an algorithm that determines the optimal supply chain configuration that minimizes the total supply chain cost. Second, we want to provide general insights and conditions on when certain supply chain structures are appropriate.

3.2 Stage Notation and Assumptions[1]

3.2.1 Option definition

An option at a stage is defined as a {direct cost added, production lead-time} pairing. There are O_i options to choose from at stage i and we define the jth available option by O_{ij} = {c_{ij} , T_{ij}} where c_{ij} denotes the direct cost added and T_{ij} denotes the production lead-time of the jth option at stage i; note that $1 \le j \le O_i$. When a stage reorders, the production lead-time is the time to process an item at the stage, assuming all of the inputs are available. The production lead-time includes both the waiting and processing time at the stage, plus any transportation time required to put the item into inventory. For instance, suppose stage i's selected option has a three-day production lead-time. If we make a production request on stage i in time period t, then stage i completes the production at time t+3, provided that there is an adequate supply from stage i-1 at time t.

Since we assume that the production lead-time is not impacted by the size of the order, we in effect assume that there are no capacity constraints that limit production at a stage.

An option's direct cost represents the direct material and direct labor costs associated with the option. If the option is the procurement of a raw material from a vendor then the direct costs would be the purchase price and the labor cost to unpack and inspect the product.

3.2.2 Periodic-review base-stock replenishment policy

We assume that all stages operate according to a periodic review policy with a common review period. Each period each stage observes demand either from an external

[1] Section 3.2 is modeled after Graves and Willems (1998).

customer or from its downstream stage, and places an order on its supplier to replenish the observed demand. In effect, each stage operates with a one-for-one or base-stock replenishment policy. There is no time delay in ordering; hence, in each period the ordering policy passes the external customer demand back up the supply chain so that all stages see the customer demand.

3.2.3 External demand

Without loss of generality, we assume that external demand occurs only at node N. Let $d_N(t)$ denote the demand at stage N in period t. We assume that the demand for the end item comes from a stationary process for which the average demand per time period is μ_N. We also assume that the demand process is bounded by the function $D_N(\tau)$, for $\tau = 1, 2, 3, \ldots M_N$, where M_N is the maximum possible replenishment time for the item. That is, $D_N(\tau) \geq d_N(t) + d_N(t+1) + \ldots + d_N(t+\tau-1)$ for all t and for $\tau = 1, 2, 3, \ldots M_N$. We define $D_N(0) = 0$ and assume that $D_N(\tau)$ is increasing and concave on $\tau = d_i(t) = \phi_{i,i+1} d_{i+1}(t)$, 2, 3, ... M_N; thus, $D_N(\tau) - D_N(\tau-1)$ is nonnegative and decreases as τ increases.

3.2.4 Internal demand

We term an internal stage to be one with internal customers or successors. In the serial line formulation, these are stages with labels 1, 2, …, N-1. For an internal stage, the demand at time t equals the order placed by its immediate successor. Since each stage orders according to a base stock policy, the demand at internal stage i is denoted as $d_i(t)$ and given by:

$$d_i(t) = \phi_{i,i+1} d_{i+1}(t)$$

for i = 1, 2, …, N-1 where $\phi_{i,i+1}$ denotes the number of units of stage's i's product necessary to produce one unit of stage i+1's product.

We assume that the demand at each internal node of the supply chain is also stationary and bounded. The average demand rate for component i is:

$$\mu_i = \phi_{i,i+1} \mu_{i+1}.$$

We also assume that demand for the component i is bounded by the function $D_i(\tau)$, for τ = 1, 2, 3, ... M_i, where M_i is the maximum possible replenishment time for the item. For the serial case, the demand bound at stage i is derived directly from the demand bound at stage i+1.

3.2.5 Guaranteed service times

We assume that the demand node N promises a guaranteed service time S_N by which the stage will satisfy customer demand. For instance, if $S_N = 0$, then the stage provides immediate service from inventory to the final customer; if $S_N > 0$, then the customer demand at time t, $d_N(t)$, must be filled by time $t + S_N$. Furthermore, we assume that stage N provides 100% service for the specified service time: stage N delivers exactly $d_N(t)$ to the customer at time $t + S_N$. These guaranteed service times for the end items are model inputs.

An internal stage i quotes and guarantees a service time S_i to its downstream stage i+1. Given the assumption of a base-stock policy, stage i+1 places an order equal to $\phi_{i,i+1} d_{i+1}(t)$

on stage i at time t; then stage i delivers exactly this amount to stage i+1 at time $t + S_i$. For instance, if $S_i = 3$, then stage i will fulfill at time t + 3 an order placed at time t by stage i+1. These internal service times are decision variables for the optimization model, as will be seen.

3.3 Single-Stage Single-Option Model[2]

In this section, we present a model for the inventory at a single stage, where there is only one option available at the stage. The single-stage model serves as the building block for modeling a multi-stage supply chain. Since we assume that there is only one option per stage, we can suppress the option-specific index and denote the production lead-time at stage i by T_i.

We have already noted that each stage quotes and guarantees a service time S_i by which stage i will deliver product to its immediate successor. For a serial supply chain, it must also be the case that stage i is being quoted a service time by its upstream supplier. That is, the inbound service time to stage i is the service time that stage i-1 quotes to stage i. By definition, this inbound service time is equal to S_{i-1}. For the case where i = 1, we assume that $S_0 = 0$; this corresponds to the case where there is an infinite supply of material available to the supply chain.

3.3.1 Inventory model

We define $I_i(t)$ to be the finished inventory at stage i at the end of period t, where we assume the inventory system starts at time t=0. Under the assumptions of perfect service and a base-stock replenishment policy, we can express $I_i(t)$ as

$$I_i(t) = B_i - d_i(t - S_{i-1} - T_i, t - S_i) \quad (1)$$

where $B_i = I_i(0) \geq 0$ denotes the base stock and where $d_i(a, b)$ denotes the demand at stage i over the time interval (a, b]. [see Kimball 1955, Simpson 1958 or Graves 1988] Since we assume a periodic-review replenishment policy, then without loss of generality we express all time parameters as integer units of the underlying time period. Hence, we understand $d_i(a, b)$, the demand at stage i over the time interval (a, b], to be given by

$$d_i(a, b) = d_i(a+1) + d_i(a+2) + \ldots + d_i(b)$$

for $a<b$ and $d_i(t)$ being the demand observed at stage i in time period t. When $a \geq b$, we define $d_i(a, b) = 0$. And for (1) to be true for small t, we define $d_i(a, b) = d_i(0, b)$ for $a<0$.

To explain (1), we observe that the replenishment time for the inventory at stage i is $S_{i-1} + T_i$. Thus, in time period t stage i completes the replenishment of the demand observed in time period $t - S_{i-1} - T_i$. Hence, at the end of time period t, the cumulative replenishment to the inventory at stage i equals $d_i(0, t - S_{i-1} - T_i)$. For a given service time S_i, in time period t stage i fills the demand observed in time period $t - S_i$ from its inventory. By the end of time period t the cumulative shipments from the inventory at stage i equal $d_i(0, t - S_i)$. The difference between the cumulative replenishment and the cumulative shipments is the inventory shortfall, $d_i(t - S_{i-1} - T_i, t - S_i)$. The on-hand inventory at stage i is the initial inventory or base stock minus the inventory shortfall, as given by (1).

---

[2] Sections 3.3.1 to 3.3.4 are modeled after Graves and Willems (1998).

3.3.2 Determination of base stock

We require that $I_i(t) \geq 0$ with probability 1 in order for the stage to provide 100% service to its customers. From (1) we see that 100% service requires that

$$B_i \geq d_i(t - S_{i-1} - T_i, t - S_i) \quad \text{with probability 1.}$$

Since we assume demand is bounded, we can satisfy the above requirement with the least inventory by setting the base stock as follows:

$$B_i = D_i(\tau) \quad \text{where } \tau = \max\{0, S_{i-1} + T_i - S_i\}. \qquad (2)$$

By assumption, any smaller value for the base stock can not assure that $I_i(t) \geq 0$ with probability 1, and thus cannot guarantee 100% service.

In words, we set the base stock equal to the maximum possible demand over the *net* replenishment time for the stage. The replenishment time for stage i is the time to get the inputs (S_{i-1}) plus the production time at stage i (T_i). The *net* replenishment time for stage i is the replenishment time minus the service time (S_i) quoted by the stage. The demand over the net replenishment time is demand that has been filled but that has not yet been replenished. The base stock must cover this time interval of exposure; thus the base stock is set to the maximum demand over this time interval.

It is possible that the promised service time is longer than the replenishment time, i.e., $S_{i-1} + T_i < S_i$, and thus the net replenishment time is negative. For example, it may take five days for the stage to replenish its inventory, but the promised service time is eight days. In this case, we see from (2) that there is no need for a finished goods inventory; we can set the base stock B_i to zero and still provide 100% service. Indeed, in such a case, the stage would delay each order on its suppliers by $S_i - S_{i-1} - T_i$ periods, so that the supplies arrive when needed.

With no loss of generality, we can redefine the inbound service time so that the net replenishment time is nonnegative. In particular we redefine S_{i-1} to be the smallest value that satisfies the following constraints:

$$S_{i-1} \geq S_i \quad \text{for } i = 1, 2, \ldots, N \quad \text{and}$$

$$S_{i-1} + T_i \geq S_i \quad .$$

If the inbound service time is such that $S_{i-1} > S_i$ for some i = 1, 2, …, N, then stage i delays orders from stage i by S_{i-1} - S_i periods.

3.3.3 Safety stock model

We use (1) and (2) to find the expected inventory level $E[I_i]$:

$$E[I_i] = B_i - E[d_i(t - S_{i-1} - T_i, t - S_i)]$$

$$= D_i(S_{i-1} + T_i - S_i) - (S_{i-1} + T_i - S_i)\, \mu_i \qquad (3)$$

for $S_{i-1} + T_i - S_i \geq 0$. The expected inventory represents the safety stock held at stage i. The safety stock is a function of the net replenishment time and the bound on the demand process.

3.3.4 Pipeline Inventory

In addition to the safety stock, we want to account for the in-process or pipeline stock at the stage. Following the argument for the development for equation (1), we observe that the work-in-process inventory at time t is given by

$$W_i(t) = d_i(t - S_{i-1} - T_i, t - S_{i-1}) .$$

That is, the work-in-process corresponds to T_i periods of demand given the assumption of a deterministic production lead-time for the stage. The amount of inventory on order at time t is

$$O_i(t) = d_i(t - S_{i-1}, t)$$

where the units of $O_i(t)$ are in terms of finished items at stage i, and denote the amount of component kits on order from stage i-1 to stage i. Similar to the work-in-process, the amount on order equals S_{i-1} periods of demand.

From (1) we see that the finished inventory plus the work-in-process plus the on-order inventory is a constant, namely the base stock. Furthermore, we see that the expected work-in-process depends only on the lead-time at stage i and is not a function of the service times:

$$E[W_i] = T_i \mu_i . \quad (4)$$

3.3.5 Safety stock cost calculation

To determine the safety stock cost at stage i, we need to determine stage i's holding cost. Since this section is considering only the single-option model for each stage we know the direct costs added at the stage is c_i. For the purposes of calculating holding costs, it is necessary to determine the total direct costs that have been added from stage 1 up to and including the current stage. For stage i, denote C_i as the total direct cost added up to and including stage i; i.e., the cumulative cost at stage i. Since the supply chain is a serial line by assumption, C_i is determined by the trivial recursion $C_i = C_{i-1} + c_i$ for stages i = 1, …, N and $C_0 = 0$.

If we assume a holding cost rate of α then the per unit holding cost at stage i equals αC_i. Therefore, the expected safety stock cost at stage i equals $\alpha C_i E[I_i]$.

3.3.6 Pipeline stock cost calculation

To determine the cost of the pipeline stock at stage i, we multiply the pipeline stock at the stage by the average value of the stock at the stage. If we assume that the costs accrue as a linear function of the time spent at the stage, then the average value of a unit of pipeline stock at stage i equals $(C_{i-1} + C_i)/2$. This can also be written as $C_{i-1} + c_i/2$. Therefore, the expected pipeline stock cost at stage i equals $\alpha(C_{i-1} + c_i/2)E[W_i]$. This assumption of a linear cost-accrual process approximates the real process; it can be changed to a more complicated function if conditions justify this.

3.4 Multi-Stage Multi-Option Serial Supply Chain Model

This section uses Section 3.3 as the building block in order to model the expected safety stock levels and pipeline stock levels across the serial supply chain. The consideration of multiple options at a stage does introduce some additional complexity to the formulation. In particular, we need to explicitly account for the fact that only one option will be selected at each stage. To do this, we introduce a 0-1 indicator variable y_{ij} for i = 1, 2, …, N and $1 \leq j \leq$

O_i. y_{ij} equals 1 if option j is selected for stage i and equals 0 otherwise; i.e., $y_{ij} = 1$ implies O_{ij} is selected. Given this additional notation, we can formulate the model for stage i as:

$$E[I_i] = \sum_{j=1}^{O_i} y_{ij}\left[D_i\left(S_{i-1} + T_{ij} - S_i\right) - \left(S_{i-1} + T_{ij} - S_i\right)\mu_i\right] \tag{5}$$

$$E[W_i] = \sum_{j=1}^{O_i} y_{ij} T_{ij} \mu_i \tag{6}$$

$$y_{ij}\left(S_{i-1} + T_{ij} - S_i\right) \geq 0 \qquad \text{for } 1 \leq j \leq O_i \tag{7}$$

$$\sum_{j=1}^{O_i} y_{ij} = 1 \tag{8}$$

$$y_{ij} \in \{0,1\} \qquad \text{for } 1 \leq j \leq O_i \tag{9}$$

Equation (5) expresses the expected safety stock as a function of the net replenishment time and demand characterization given that option j is selected at stage i. In (6), the expected pipeline stock equals the mean demand times the selected option's production time. Equation (7) ensures that the net replenishment time is nonnegative. Finally, equations (8) and (9) require that exactly one option be selected at each stage.

We see from (5)-(9) that the expected inventory in the supply chain is a function of the demand process, the options selected and the service times. We assume that the options' production lead-times, and the means and bounds of the demand processes are known input parameters. The guaranteed service time for stage N is also an input. Thus, in any optimization context, the internal service times and options selected are the decision variables.

In order to determine which options and service times are optimal, we need to know how choosing a particular option and service time configuration affects the supply chain's costs. This is the subject of Section 3.5.

3.5 Multi-Stage Multi-Option Objective Function Determination

This section presents the formulation of the supply chain costs that are relevant to the supply chain configuration problem. There are three relevant costs: manufacturing cost, safety stock cost, and pipeline stock cost. Safety stock and pipeline stock costs have already been introduced and only need to be modified to handle the addition of multiple options at a stage. Cost of goods sold equals the total direct cost of all the units of product that are shipped to consumers.

Note that all three of these costs are influenced by the option chosen at each stage. For example, a supply chain comprised of stages with low direct costs and long lead times may have a low cost of goods sold but a high safety stock cost. To make this statement more rigorous, we must first know the direct and cumulative cost added at each stage. In particular, our calculation of C_i must take the selected options into account. This is done as follows:

$$C_i = C_{i-1} + \sum_{j=1}^{O_i} y_{ij} c_{ij} \tag{10}$$

for i = 1, 2, …, N where $C_0 = 0$. With this cost information, we can now determine the supply chain's cost.

3.5.1 Safety Stock Cost

By definition, safety stock is held at the end of the stage, after its processing activity has occurred. Therefore, the value of a unit of safety stock at stage i is equal to the cumulative cost of the product at stage i. The expected safety stock cost at stage i is:

$$\alpha C_i E[I_i] \tag{11}$$

where α represents the holding cost rate.

3.5.2 Pipeline Stock Cost

To determine the cost of the pipeline stock for stage i, we multiply the expected pipeline stock by the average cost of the product at the stage. Two equivalent cost calculations are shown below:

$$\left(C_{i-1} + \frac{\sum_{j=1}^{O_i} y_{ij} c_{ij}}{2} \right) \alpha E[W_i] = \left(C_i - \frac{\sum_{j=1}^{O_i} y_{ij} c_{ij}}{2} \right) \alpha E[W_i] \tag{12}$$

3.5.3 Cost of Goods Sold

Cost of goods sold (COGS) represents the total cost of all the units that are delivered to customers during a company-defined interval of time. Typically, the interval of time is one year. The cost of goods sold is determined by multiplying the end item's annual demand times the end item's unit manufacturing cost. That is,

$$COGS = \beta C_N \mu_N \tag{13}$$

where β is a scalar that converts the model's underlying time unit into the company's time interval of interest; β is the scalar that expresses (13) in the same units as (11) and (12). Recall that the model has an underlying time unit that is common to all stages. For example, if the model's underlying time unit is one day and the company's interval of interest is one year, then we would need to multiply μ_N by 365 to get the expected annual volume of the product. This annual volume would then be multiplied by the unit manufacturing cost, C_N, to get the expected cost of goods sold per year.

The above derivation of COGS is formulated from the perspective of the end item. For an intuitive understanding of the cost, this is an easier interpretation. However, when formulating the objective function, we will find it useful to divide the cost among the stages in the supply chain. To do this, we note that the cumulative cost at stage N is just the summation of the chosen direct costs at each of the stages. Therefore, we can calculate COGS as follows:

$$\text{COGS} = \beta \sum_{i=1}^{N} \sum_{j=1}^{O_i} y_{ij} c_{ij} \mu_i \qquad (14)$$

This formulation is analogous to the echelon stock cost seen in many classic multiechelon inventory works, including Clark and Scarf (1960).

3.6 Math Programming Formulation

With the inventory calculations in Section 3.4 and the cost calculations in Section 3.5, we are now in a position to formulate an optimization problem for finding the optimal option and service time configuration for the entire serial supply chain:

P

$$\min \sum_{i=1}^{N} \sum_{j=1}^{O_i} y_{ij} \Big[\alpha C_i \Big[D_i \left(S_{i-1} + T_{ij} - S_i \right) - \left(S_{i-1} + T_{ij} - S_i \right) \mu_i \Big] + \left(C_i - \frac{c_{ij}}{2} \right) \alpha T_{ij} \mu_i + \beta c_{ij} \mu_i \Big]$$

st

$$C_i - C_{i-1} - \sum_{j=1}^{O_i} y_{ij} c_{ij} = 0 \qquad \text{for } i = 1,2,\ldots,N$$

$$y_{ij}\left(S_{i-1} + T_{ij} - S_i\right) \geq 0 \qquad \text{for } i = 1,2,\ldots,N, 1 \leq j \leq O_i$$

$$S_N \leq s_N$$

$$\sum_{j=1}^{O_i} y_{ij} = 1 \qquad \text{for } i = 1,2,\ldots,N$$

$$y_{ij} \in \{0,1\} \qquad \text{for } i = 1,2,\ldots,N, 1 \leq j \leq O_i$$

$$S_i \geq 0 \text{ and integer} \qquad \text{for } i = 1,2,\ldots,N$$

where s_N is the guaranteed service time for demand node N; s_N is a user-specified input to the model. Thus, the objective of problem **P** is to minimize the sum of the supply chain's safety stock cost, pipeline stock cost, and cost of goods sold. The constraints assure that exactly one option is chosen per stage, that the net replenishment time for each stage is nonnegative and that Stage N satisfies its service guarantee. The decision variables are the service times and the options selected.

Problem **P** is an integer nonlinear optimization problem. For a fixed set of feasible y_{ij} (corresponding to the case where the user specifies the option selected at each stage), Graves and Willems (1998) show that the objective function is a concave function provided that the demand bound $D_i(\ )$ is a concave function for each stage i. Hence, in the single-option case, we minimize a concave function over a set of linear constraints. Although the feasible region is not necessarily bounded, one can show that the optimal service times need not exceed the sum of the production lead-times, provided that the demand bound $D_i(\ )$ is a non-decreasing function for each stage i. Thus, the problem for this restricted version of **P** is to minimize a concave function over a closed, bounded convex set. An optimum for such problems is at an extreme point of the feasible region (e.g., Luenberger, 1973).

3.7 Dynamic Programming Solution Procedure

The serial line case can be solved to optimality by dynamic programming. Section 3.7.1 constructs the dynamic program's state space and section 3.7.2 provides the solution procedure.

3.7.1 State space determination

In order to solve the dynamic program efficiently, we need to define a state space that allows the algorithm to solve the network in a node-by-node fashion, using only information that is locally available at the node. When there is only one available option per stage, Graves and Willems (1998) show how to formulate the dynamic program with a single state variable. The state variable is either the inbound or outbound service time at the stage. The type of service time that is used at a stage depends on the where the stage resides in the network.

The single-option problem only requires one state variable because several key parameters are uniquely determined by the options. In particular, the maximum replenishment time and the cumulative cost at each stage are known constants if there is only one available option per stage. Having a constant cumulative cost is important because this makes the pipeline stock and cost of goods sold deterministic quantities. These two costs do not depend on service times so the options chosen entirely determine their values. Therefore, when there is only one available option per stage, the optimization problem simplifies to determining the optimal set of service times that minimize the supply chain's safety stock cost.

The multi-option serial supply chain problem can be modeled using two state variables. As in Graves and Willems (1998), one state variable will represent the outbound service time at the stage. The additional state variable will be the cumulative cost at the stage.

We define the set of feasible cumulative costs at stage i by X_i. Since the cumulative cost at stage i is determined by the options selected at stages 1 to i, and there are a finite number of options at each stage, the cumulative cost at stage i can only take on a set of discrete values. For example, if stage 1 has two options then it can have at most two cumulative costs; in this case, each possible cumulative cost is equal to the direct cost for one of the options. If stage 2 also has two options, then stage 2 can have at most four cumulative costs; these are created by adding each option's cost element to stage 1's cumulative costs.

We will also find it useful to define XI_i as the set of incoming cumulative costs to stage i. For the serial line supply chain, these are just the cumulative costs at stage i-1. In set notation, $XI_i \subset X_{i-1}$.

3.7.2 Forward recursive formulation

The dynamic program is a forward recursion starting at stage 1 and proceeding to stage N. For each stage, the dynamic program evaluates a functional equation denoted by $f_i(C, S)$. $f_i(C, S)$ is defined as the minimum supply chain cost for node 1 to i given that stage i's cumulative cost is C and stage i quotes a service time of S.

To develop the functional equation, we first define the supply chain cost at stage i as a function of the service time quoted to stage i (SI), plus stage i's service time (S), cumulative cost (C) and option (O_{ij}) selected:

$$g_{ij}(SI, C, S) = \alpha C\left[D_i(SI + T_{ij} - S) - (SI + T_{ij} - S)\mu_i\right] + \left(C - \frac{c_{ij}}{2}\right)\alpha T_{ij}\mu_i + \beta c_{ij}\mu_i \quad (15)$$

$g_{ij}(SI, C, S)$ is the summation of the safety stock cost, pipeline stock cost, and direct manufacturing cost contributed by the stage. By observation, $g_{ij}(SI, C, S)$ is strictly decreasing in S over the interval $0 \le S \le SI + T_{ij}$ and strictly increasing in SI over the interval $[S - T_{ij}]^+ \le SI \le M_{i-1}$.

There are three conditions on $g_{ij}(SI, C, S)$; one condition corresponding to each of the function's parameters. First, we require $[S - T_{ij}]^+ \le SI \le M_{i-1}$. The left inequality constrains the service time quoted to stage i (SI) so that the net replenishment time at stage i is nonnegative. The right inequality restricts the service time at stage i-1 to not exceed the maximum service time that stage i-1 can quote. The second condition is that $0 \le S \le SI + T_{ij}$. The service time at stage i must be nonnegative and can not exceed the net replenishment time. Third, we require that the incoming cumulative cost to stage i equal a cost that the upstream configuration can produce. This requires the cumulative cost (C) minus option j's direct cost (c_{ij}) to equal a feasible cumulative cost at stage i-1. Thus, for C to be feasible, we must have $\{C - c_{ij}\} \in X_{i-1}$.

We now want to define the minimum supply chain cost for stages 1 through i given that stage i utilizes option O_{ij}. Let $f_{ij}(C, S)$ denote this option-specific optimal cost-to-go function. It is defined below:

$$f_{ij}(C, S) = \min_{SI}\left\{g_{ij}(SI, C, S) + f_{i-1}(C - c_{ij}, SI)\right\} \tag{16}$$

The first term represents the supply chain's costs incurred at stage i and is defined in (15). The second term corresponds to the minimum cost for the stages that are upstream of stage i. For these upstream stages, we include their minimum supply chain costs as a function of stage i-1's service time, SI, and its cumulative cost $C - c_{ij}$. The three conditions on (15) also apply to (16).

The functional equation for $f_i(C, S)$ is:

$$f_i(C, S) = \min_{j}\left(f_{ij}(C, S)\right) \tag{17}$$

where the minimization is over the options at stage i that are feasible given a cumulative cost of C at stage i and a service time of S.

The functional equation is evaluated for all {cumulative cost, service time} states that are feasible at stage i. Thus, for each $C \in X_i$, we solve for $S = 0, 1, \ldots, M_i$. M_i, the maximum replenishment time at stage i, is calculated by the recursion $M_i = M_{i-1} + \max\{T_{ij}\}$. The set X_i is defined by the recursion $X_i = \{x_{i-1} + c_{ij} \mid x_{i-1} \in X_{i-1}, j = 1, \ldots, O_i\}$.

To find the optimal solution, we first note that the service time at stage N can not exceed s_N. Therefore, for each feasible cumulative cost C at stage N, we can evaluate $f_N(C, s_N)$ and choose the option with the minimum cost. We can then backtrack through the network to produce the optimal option and service time at each stage.

4 Assembly Network Formulation

4.1 Network Representation

A supply chain that can be modeled as an assembly network is one in which each stage can receive inputs from several adjacent suppliers but can directly supply only one downstream stage. In network terms, an assembly network is a graph where each node can have multiple incoming arcs but only one outgoing arc. We assume the nodes are topologically ordered. That is, for every arc $(i,j) \in A$, $i < j$. By construction, this implies that the finished goods node will be labeled node N.

Let B(i) denote the set of stages that are backwards adjacent to stage i; $B(i)=\{h:(h,i)\in A\}$. The cardinality of B(i), denoted $|B(i)|$, equals the number of stages directly supplying stage i.

For each node i we define N_i to be the subset of nodes {1, 2, ... i} that are connected to i on the sub-graph consisting of nodes {1, 2, … i}. That is, N_i is the set of nodes that form an in-tree rooted at node i. We will use N_i to explain the dynamic programming recursion later in this section. We can determine N_i by the following equation:

$$N_i = \{i\} + \bigcup_{h \in B(i)} N_h .$$

An example of a typical supply chain is shown in Figure 4-1:

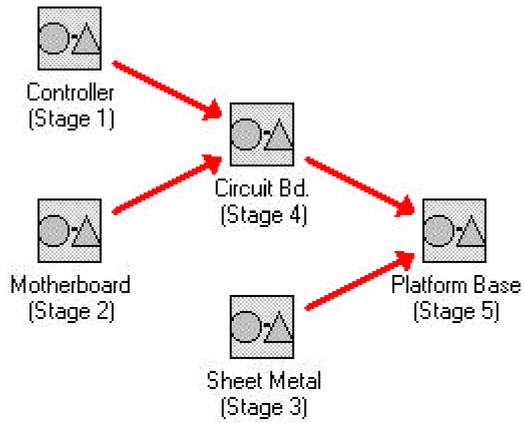

Figure 4-1: Assembly network supply chain

and the options at each stage are shown in Table 4-1:

Stage	Option	Description	Direct Cost	Lead-time
1	1	Multinational Supplier	$5	10 days
1	2	Local Supplier	$10	4 days
2	1	Multinational Supplier	$5	10 days
2	2	Local Supplier	$10	4 days
3	1	Local Vendor #1	$15	20 days
3	2	Local Vendor #2	$20	10 days
4	1	Manual Assembly	$25	10 days
4	2	Automated Assembly	$30	8 days
5	1	Low Volume Equipment	$10	30 days
5	2	High Volume Equipment	$15	15 days

Table 4-1: Options available for assembly network supply chain

Figure 4-1 represents a stylized supply chain for a subassembly that is created by inserting a circuit board into a metal housing. The circuit board has two main components, a motherboard and a controller. All of the stages have two sourcing options, consisting of a low cost, long lead-time supplier and a higher cost, shorter lead time supplier.

In Figure 4-1, N_i is {3} for i=3 and {1, 2, 4} for i=4.

4.2 Stage Notation and Assumptions

The assumptions and notation adopted for the serial network are still valid for assembly networks. This section addresses two differences between the serial line and assembly network cases. First, the notation for the demand process has to be redefined now that the network is not a serial line. Second, the incoming service time to a stage has to be defined since a stage can have several upstream suppliers, each quoting the stage a different service time.

4.2.1 Internal demand

For an internal stage, the demand at time t equals the order placed by its immediate successor. Since each stage orders according to a base stock policy, the demand at internal stage i is denoted as $d_i(t)$ and given by:

$$d_i(t) = \phi_{ij} d_j(t)$$

where ϕ_{ij} denotes the number of units of stage's i's product necessary to produce one unit of stage j's product.

We assume that the demand at each internal node of the supply chain is also stationary and bounded. The average demand rate for component i is:

$$\mu_i = \phi_{ij} \mu_j .$$

We also assume that demand for the component i is bounded by the function $D_i(\tau)$, for $\tau = 1, 2, 3, \ldots M_i$, where M_i is the maximum possible replenishment time for the item. For the assembly case, the demand bound at stage i is derived from the demand bound at its downstream adjacent stage j.

4.2.2 Guaranteed service times

Since each stage has only one downstream customer, S_i still represents the service time that stage i quotes to its downstream customer. However, the possibility of multiple upstream adjacent stages requires additional notation to characterize the incoming service time quoted to a stage. Let SI_i denote the maximum incoming service time quoted to stage i. That is, $SI_i = \max \{ S_h \}$ for all h such that $(h,i) \in A$. This assumes that stage i must wait until all of its raw materials arrive before it can begin its processing function.

4.3 Solution Procedure

4.3.1 Dynamic programming formulation

As in the serial line formulation, the state variables for the assembly network formulation are service time and cumulative cost. However, the assembly case is complicated by the fact that different configurations at upstream stages can produce the identical cumulative cost at the downstream stage. Since these different configurations will have different supply chain costs, we need a way to efficiently enumerate and evaluate these configurations in order to determine the optimal cost-to-go for the downstream stage. Therefore, before the dynamic programming algorithm can be presented, a new data structure must first be created.

4.3.1.1 Incoming cumulative cost combinations

Let CI_i denote the incoming cumulative cost to stage i. This is equal to the cumulative cost at stage i, C_i, minus the stage's direct cost added, c_{ij}. For the assembly network case, we need a data structure that allocates CI_i across the stages in B(i). In the serial network case, the allocation is immediate; $CI_i = C_{i-1}$ since B(i) = {i-1}. In the assembly network case, we define a *combination* at stage i as a set comprised of |B(i)| elements, each element corresponding to a feasible cumulative cost for one of the stages in B(i). Summing the elements of the combination will equal the incoming cumulative cost CI_i.

Let $Q_i(CI)$ denote the set of combinations where the summation of each combination equals CI. For a combination $q \in Q_i(CI)$, we define v_{qh} as the cumulative cost at stage h associated with combination q; that is, $v_{qh} \in X_h$.

For example, in Figure 4-1, two combinations produce an incoming cumulative cost of \$65 at stage 5. The configurations are {\$45, \$20} and {\$50, \$15} where the first term of each combination is the cumulative cost at stage 3 and the second term is the cumulative cost at stage 4. In the notation above: B(5) = {3, 4}; $|Q_5(\$65)| = 2$; $Q_5(\$65)$ = { {\$45, \$20}, {\$50, \$15} }; v_{13} = {\$45}, v_{14} = {\$20} and v_{23} = {\$50} , v_{24} = {\$15}.

4.3.1.2 Forward recursive formulation

The dynamic program is a forward recursion, starting at stage 1 and proceeding to stage N. For each stage, the dynamic program evaluates a functional equation denoted by $f_i(C, S)$. $f_i(C, S)$ is defined as the minimum supply chain cost for the in-tree rooted at node i given that stage i has a cumulative cost C and quotes a service time of S.

To develop the functional equation, we first define the supply chain cost for stage i as a function of the maximum service time quoted to stage i (SI), plus stage i's service time (S), cumulative cost (C) and option selected (O_{ij}):

$$g_{ij}(SI, C, S) = \alpha C\left[D_i(SI + T_{ij} - S) - (SI + T_{ij} - S)\mu_i\right] + \left(C - \frac{c_{ij}}{2}\right)\alpha T_{ij}\mu_i + \beta c_{ij}\mu_i \quad (18)$$

Note that (18) is exactly the same as (15). It is only included here for completeness.

We next want to characterize the minimum total supply chain cost for each of the sub-networks that are upstream of stage i. That is, we want to calculate the total cost-to-go for each subnetwork N_h, where $h \in B(i)$. Let $FI_i(CI,SI)$ denote the minimum total upstream cost-to-go given that the incoming cumulative cost to stage i is CI and the maximum incoming service time to stage i is SI. $FI_i(CI,SI)$ is defined below:

$$FI_i(CI, SI) = \min_{q \in Q_i(CI)} \left\{ \sum_{h \in B(i)} f_h(v_{qh}, SI) \right\} \quad (19)$$

Equation (19) finds the minimum total supply chain cost for the in-trees rooted at stage i's upstream adjacent stages. For a given combination q, the function loops over all of the upstream adjacent stages and returns the minimum cost-to-go for each stage given the maximum service time it can quote and its allocated portion of stage i's incoming cumulative cost. The summation of these |B(i)| terms equals the cost of the combination quoting a maximum service time of SI. To find the minimum total supply chain cost, we must minimize over all upstream combinations belonging to the set $Q_i(CI)$.

We now define the minimum cost-to-go at stage i given that stage i utilizes option O_{ij}.

Let $f_{ij}(C, S)$ denote this option-specific optimal cost-to-go function. It is defined below:

$$f_{ij}(C, S) = \min_{SI} \left\{ g_{ij}(SI, C, S) + FI_i(C - c_{ij}, SI) \right\} \quad (20)$$

The first term represents the supply chain costs incurred at stage i and is defined in (18). The second term, defined in (19), represents the minimum total supply chain cost for the subgraph that is upstream adjacent to stage i. Since the cumulative cost at stage i is C, this subgraph's cumulative cost must equal $C - c_{ij}$.

We can now use (20) to develop the functional equation for $f_i(C, S)$:

$$f_i(C,S) = \min_j \{f_{ij}(C,S)\} \tag{21}$$

where the minimization is over the available options at stage i. The minimization can be done by enumeration.

The functional equation is evaluated for all {cumulative cost, service time} states that are feasible at stage i. Thus, for each $C \in X_i$, we solve for S = 0, 1, …, M_i. M_i, the maximum replenishment time at stage i, is calculated by the recursion

$$M_i = \max_{h \in B(i)} \{M_h\} + \max_{1 \le j \le O_i} \{T_{ij}\}.$$

To find the optimal solution, we first note that the service time at stage N can not exceed s_N. Therefore, for each feasible cumulative cost C at stage N, we can evaluate $f_N(C, s_N)$ and choose the option with the minimum cost. We can then backtrack through the network to produce the optimal option and service time at each stage.

5 Distribution Network Formulation

5.1 Network Representation

A supply chain that can be modeled as a distribution network is one in which each stage can have only one supplier and one or more customers. In network terms, a distribution network is a graph where each node can have multiple outgoing arcs but only one incoming arc. We assume the nodes are topologically ordered. That is, for every arc $(i,j) \in A$, $i < j$. By construction, this implies that the raw material node will be labeled node 1.

Let D(i) denote the set of stages that are forward adjacent to stage i; $D(i)=\{k:(i,k)\in A\}$. The cardinality of D(i), denoted |D(i)|, equals the number of stages directly served by stage i.

For each node i we define N_i to be the subset of nodes {i, i+1, ... N} that are connected to i on the sub-graph consisting of nodes {i, i+1, ... N}. We will use N_i to explain the dynamic programming recursion. We can determine N_i by the following equation:

$$N_i = \{i\} + \bigcup_{k \in D(i)} N_k \, .$$

An example of a typical supply chain is shown in Figure 5-1:

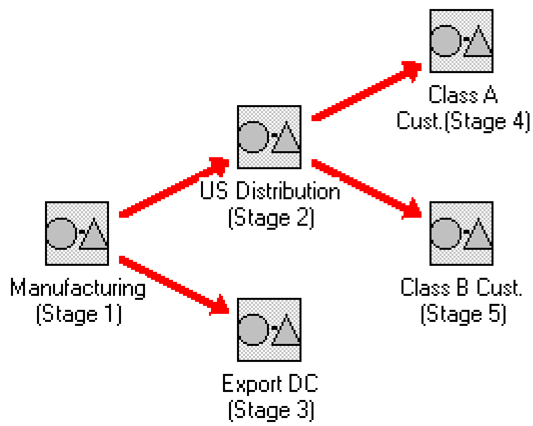

Figure 5-1: Distribution network supply chain

and the options at each stage are shown in Table 5-1:

Stage	Option	Description	Direct Cost	Lead-time
1	1	Low Volume Equipment	$10	30 days
1	2	High Volume Equipment	$15	15 days
2	1	3rd Party Carrier	$3	5
2	2	Premium Carrier	$6	2
3	1	Shipment by Boat	$5	30 days
3	2	Shipment by Air	$25	3 days
4	1	3rd Party Carrier	$6	10 days
4	2	Premium Carrier	$12	3 days
5	1	3rd Party Carrier	$6	10 days
5	2	Premium Carrier	$12	3 days

Table 5-1: Options available for distribution network supply chain

Figure 5-1 represents a stylized supply chain for a product's distribution system. The product serves both a domestic and export market. For the domestic market, there are two classes of customers. All of the stages have two sourcing options, consisting of premium and basic transportation vendors.

In Figure 5-1, N_i is {3} for i=3 and {2, 4 ,5} for i=2.

5.2 Additional Stage Assumptions

The assumptions and notation adopted for the serial network are still valid for distribution networks. However, the demand process and the impact on service times have to be clarified.

5.2.1 Demand assumptions[3]

5.2.1.1 External demand

We assume that the demand process for each end item behaves in the same manner as the demand process for the single end item in the serial case. We also assume that the service time for each external node is bounded. For each stage i that is an external stage, let s_i denote the maximum service time the stage can quote.

5.2.1.2 Internal demand

We term an internal stage to be one with internal customers or successors. For an internal stage, the demand at time t is the sum of the orders placed by the immediate successors. Since each stage orders according to a base-stock policy, the demand at internal stage i is given by:

$$d_i(t) = \sum_{(i,\, j) \in A} \phi_{ij}\, d_j(t)$$

where A is the arc set for the network representation of the supply chain.

We assume that the demand at each internal node of the supply chain is stationary and bounded. The average demand rate for component i is:

$$\mu_i = \sum_{(i,\, j) \in A} \phi_{ij}\, \mu_j .$$

We assume that demand for the component i is bounded by the function $D_i(\tau)$, for τ = 1, 2, 3, ... M_i, where M_i is the maximum replenishment time for the item. This bound may be

---

[3] This section is modeled after Graves and Willems (1998).

a given input or it may be derived from the demand bounds for the downstream, or customer, stages for stage i. We discuss in Graves and Willems (1998) how this might be done.

5.2.2 Service times[4]

5.2.2.1 Internal service times

An internal stage i quotes and guarantees a service time S_{ij} for each downstream stage j, $(i, j) \in A$.

For the initial development of the model, we assume that stage i quotes the same service time to all of its downstream customers; that is, we assume that $S_{ij} = S_i$ for each downstream stage j, $(i, j) \in A$. We describe in Graves and Willems (1998) how to extend the model to permit customer-specific service times. In brief, if there is more than one downstream customer, we can insert zero-cost, zero production lead-time dummy nodes between a stage and its customers to enable the stage to quote different service times to each of its customers. The stage quotes the same service time to the dummy nodes and each dummy node is free to quote any valid service time to its customer stage.

The service times for both the end items and the internal stages are decision variables for the optimization model. However, as a model input, we may impose bounds on the service times for each stage. In particular, we assume that for each end item we are given a maximum service time as an input.

[4] This section is modeled after Graves and Willems (1998).

5.3 Solution Procedure

5.3.1 Dynamic programming formulation

The state variables for the distribution network formulation are service time and cumulative cost. However, in contrast to the serial and assembly network cases, the service time state variable refers to the incoming service time quoted to the stage and the cumulative cost state variable refers to the incoming cumulative cost to the stage.

It is also worth noting that the $Q_i(C)$ data structure developed for the assembly network case is not necessary for distribution networks. Since each stage only has one upstream supplier, the option selected at the current stage uniquely determines the incoming cumulative cost to the stage.

5.3.1.1 Recursive formulation

In contrast to the previous two sections, in the distribution network the algorithm proceeds from the leaves of the network and works back towards the node with no incoming arcs. For each stage, the dynamic program evaluates a functional equation denoted by $F_i(CI, SI)$. $F_i(CI, SI)$ is defined as the minimum supply chain cost for the out-tree rooted at node i given that stage i's incoming cumulative cost is CI and stage i is quoted a service time of SI.

To develop the functional equation, we first define the supply chain cost for stage i as a function of the maximum service time quoted to stage i, plus stage i's service time, cumulative cost and option selected:

$$g_{ij}(SI, C, S) = \alpha C\left[D_i(SI + T_{ij} - S) - (SI + T_{ij} - S)\mu_i\right] + \left(C - \frac{c_{ij}}{2}\right)\alpha T_{ij}\mu_i + \beta c_{ij}\mu_i \quad (22)$$

Note that (22) is exactly the same as (15) and (18). It is only included here for completeness.

We now want to define the minimum supply chain cost for the out-tree rooted at stage i given that stage i utilizes option O_{ij}. Let $F_{ij}(C, S)$ denote this option-specific optimal cost-to-go function. It is defined below:

$$F_{ij}(CI, SI) = \min_{S} \left\{ g_{ij}(SI, CI + c_{ij}, S) + \sum_{k \in D(i)} F_k(CI + c_{ij}, S) \right\} \tag{23}$$

The first term represents the supply chain costs incurred at stage i and is defined in (22). The second term represents the minimum total supply chain cost for the subgraph that is downstream adjacent to stage i. Since the cumulative cost at stage i is $CI + c_{ij}$, the incoming cumulative cost to each of these downstream customers must equal $CI + c_{ij}$.

There are two conditions on (23). First, if stage i is an internal stage then the service time (S) must be nonnegative and it must not exceed the incoming service time (SI) plus the option's production time (T_{ij}). This condition prevents the net replenishment time from becoming negative. If stage i is an external stage, then the upper bound on S is the minimum of $SI + T_{ij}$ and s_i. Second, the incoming cumulative cost CI must be a feasible incoming cumulative cost at stage i. That is, $CI \in XI_i$.

We can now use (23) to develop the functional equation for $F_i(CI, SI)$:

$$F_i(CI, SI) = \min_{j} \left(F_{ij}(CI, SI) \right) \tag{24}$$

where the minimization is over the available options at stage i. The minimization can be done by enumeration.

The functional equation is evaluated for all {incoming cumulative cost, incoming service time} states that are feasible at stage i. Thus, for each CI $\in$ XI_i, we solve for S = 0, 1, …, $M_i - \min(T_{ij})$ for $1 \le j \le O_i$. M_i, the maximum replenishment time at stage i, is calculated by the recursion $M_i = \max_{h \in B(i)} \{M_h\} + \max_{1 \le j \le O_i} \{T_{ij}\}$.

To find the optimal solution, note that there is only one {incoming cumulative cost, incoming service time} state at stage 1; by construction this state is {$0, 0}. Therefore to find the optimal solution we just pick the option associated with $F_i(0, 0)$ and progress through the network to produce the optimal option and service time at each stage.

6 Spanning Tree Network

6.1 Network Representation

A spanning tree is a connected graph that contains N nodes and N-1 arcs. Assembly networks and distribution network are both special cases of spanning trees. Spanning trees allow the flexibility to capture numerous kinds of real world supply chains. In particular, they can model networks where a common component goes into different final assemblies that each have different distribution channels.

This section presents a generalization of the previous three sections. We again let B(i) denote the set of backwards adjacent nodes and let D(i) denote the set of forward adjacent nodes. The spanning tree uses the node labeling procedure defined in Graves and Willems (1998) and included in Appendix I. This labeling procedure differs significantly from the labeling procedures employed in previous sections. For each node i we define N_i to be the subset of nodes {1, 2, ... i} that are connected to i on the sub-graph consisting of nodes {1, 2, … i}. We will use N_i to explain the dynamic programming recursion. We can determine N_i by the following equation:

$$N_i = \{i\} + \bigcup_{\{k: k \in B(i), k < i\}} N_k + \bigcup_{\{k: k \in D(i), k < i\}} N_k \ .$$

An example of a typical supply chain is shown in Figure 6-1:

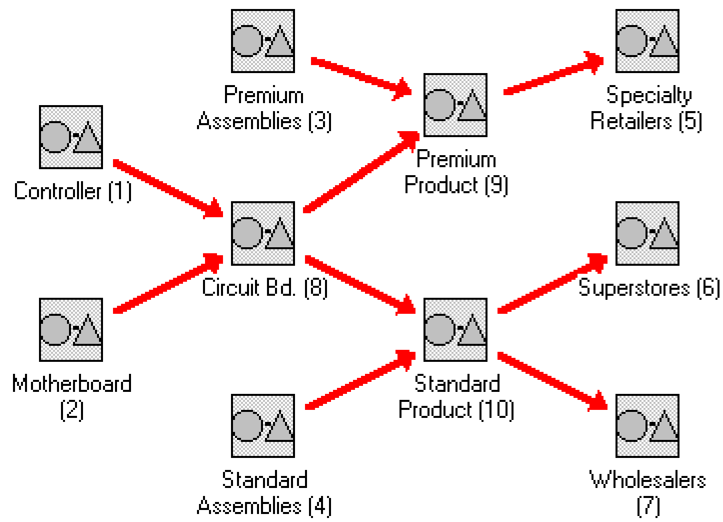

Figure 6-1: Spanning Tree network supply chain

Figure 6-1 represents a stylized supply chain for a product family that consists of two variants. The two variants share a common circuit board and are differentiated by their other assemblies. The standard product is distributed through both computer superstores and retail wholesalers. The premium product is sold only through specialty retailers.

In Figure 6-1, N_i is {1, 2, 8} for i=8 and {4, 6, 7, 8, 10} for i=10.

6.2 Stage Notation and Assumptions

The assumptions and notation adopted for the previous cases are still valid for spanning tree networks. No additional assumptions are necessary.

6.3 Solution Procedure

As with the previous network topologies, spanning tree networks can be solved as a two-state dynamic program. However, in the case of a spanning tree, there will be two forms of the functional equation, depending on the node's orientation in the network. The first form is $f_i(C, S)$, defined as the minimum supply chain cost for the subgraph N_i given that stage i has a cumulative cost C and quotes a service time of S. The second form is $F_i(CI, SI)$, defined as the minimum supply chain cost for the subgraph N_i given that stage i's incoming cumulative cost is CI and stage i is quoted a service time of SI. These two functional equations are generalizations of the functional equation for assembly networks and distribution networks, respectively.

An important property of the node labeling procedure reproduced in Appendix I is that for each node in the spanning tree, excluding the root node, there is exactly one adjacent node that has a higher label. The root node is the last node that is labeled by the labeling procedure. This adjacent node with a higher label is referred to as the parent node, and the parent node to node i is denoted p(i).

At node i for $1 \le i \le N-1$, the dynamic programming algorithm will evaluate either $f_i(C, S)$ or $F_i(CI, SI)$, depending upon on the orientation of node i relative to p(i). If p(i) is downstream of node i, then the algorithm evaluates $f_i(C, S)$. If p(i) is upstream of node i, then the algorithm evaluates $F_i(CI, SI)$. For node N, as will be seen, either functional equation can be evaluated.

6.3.1 Forward cumulative cost combination

Before the functional equations can be developed, a new data structure has to be introduced. To some extent, this data structure is an analog to the *combination* data structure that was introduced in section 4.3.1.1. In summary, the *incoming cumulative cost combination* addressed the fact that multiple upstream configurations could produce the same incoming cumulative cost at the downstream stage.

In the context of spanning trees, a similar type of situation can arise when evaluating a node with downstream adjacent stages. Recall that when solving distribution networks, if stage i supplies stage k then the incoming cumulative cost at stage k equals the outgoing cumulative cost at stage i. By the definition of a distribution network, stage i can be the only stage that supplies stage k, and hence there is a one-to-one correspondence between the upstream stage's outgoing cumulative cost and the downstream stage's incoming cumulative cost.

However, in the case of a spanning tree, a downstream stage can have more than one supplier. For example, stages i and j can both supply stage k. Therefore, when solving node i, we need to take into account that the incoming cumulative cost at stage k will not equal the outgoing cumulative cost at stage i. In fact, it is quite possible that multiple incoming cumulative costs at stage k can be associated with each outgoing cumulative cost at stage i.

Let $R_{ik}(C)$ denote the set of incoming cumulative costs at stage k that are feasible if stage i's cumulative cost is C. $R_{ik}(C)$ is defined for each $k \in D(i)$ and $C \in X_i$.

To relate the forward cumulative cost combination to the incoming cumulative cost combination, we note that $CI \in R_{ik}(C)$ implies there exists a $q \in Q_k(CI)$ such that v_{qi} equals C. Whereas the incoming cumulative cost combination acts to tie all of a stage's upstream adjacent stages together, the forward cumulative cost combination individually relates a stage to its downstream adjacent customers. This difference is due to the fact that when solving a stage involves evaluating a forward adjacent stage, all of the stages adjacent to the forward stage, besides the current stage, have already been solved. By definition of the solution procedure, a forward adjacent stage will only be evaluated if it has already been solved. And when a node is solved, it can have at most one adjacent node unsolved. Therefore, there is nothing besides the current stage being solved that relates the forward stages to one another.

6.3.2 Functional equation development

To develop the functional equation, we first define the supply chain cost for the subgraph rooted at stage i as a function of stage i's incoming service time (SI), outgoing cumulative cost (C), outgoing service time (S), and option selected (O_{ij}):

$$z_{ij}(SI,C,S) = g_{ij}(SI,C,S) + \min_{q\in Q_i(C-c_{ij})} \left\{ \sum_{\{h:h\in B(i),h<i\}} f_h(v_{qh}, SI) \right\} + \sum_{\{k:k\in D(i),k<i\}} \min_{CI\in R_{ik}(C)} \left\{ F_k(CI,S) \right\} \qquad (25)$$

The first term is the supply chain cost at stage i and has previously been discussed in section 3.7.2; it is equation (15).

The second term corresponds to the nodes in N_i that are upstream of i. The second term consists of the minimum supply chain cost for the configuration upstream of stage i, as a function of the configuration's cumulative cost and service time. The cumulative cost for the configuration is equal to the outgoing cumulative cost at stage i minus stage i's direct cost added. The incoming service time to stage i (SI) is the maximum service time that is being quoted to stage i. Therefore, SI is an upper bound on the service time that each of the upstream stages can quote. We can show that $f_h(C,SI)$, the supply chain costs for the subgraph with node set N_h, is non-increasing in the outgoing service time to node i, and thus, we equate the outgoing service time at h to the incoming service time at i without loss of generality.

The third term corresponds to the nodes in N_i that are downstream of node i. For each node k that is a customer to node i, we include the minimum supply chain cost at stage k as a function of stage i's contribution to the cumulative cost at stage k and the service time i quotes k. The argument S represents the outbound service time for node k, and thus a lower bound for the inbound service time for node k. We can show that $F_k(CI,S)$, the supply chain costs for the subgraph with node set N_k, is non-decreasing in the incoming service time to node k, and thus, we equate the incoming service time at k to the outgoing service time at i without loss of generality.

We now use the minimum supply chain cost for the subgraph with node set N_i to develop the functional equation for $f_i(C,S)$:

$$f_i(C,S) = \min_{j,SI} \left\{ z_{ij}(SI, C, S) \right\}$$

where the minimization is over the feasible set of options and incoming service times. As in the case of (15), the incoming service time is bounded by $\max(0, S - T_{ij}) \le SI \le M_i - T_{ij}$ and SI integer. This minimization can be done by enumeration.

The functional equation is evaluated for all possible integer outgoing service times and feasible cumulative costs for node i. That is, for $S = 0, 1, \ldots, M_i$ and $C \in X_i$.

The functional equation for $F_i(CI,SI)$ is of a similar structure:

$$F_i(CI, SI) = \min_{j,S} \left\{ z_{ij}(SI, CI + c_{ij}, S) \right\}$$

The minimization is over the feasible set of options and outgoing service times. If i is an internal stage, then the feasible set is $1 \le j \le O_i$ and $S = 0, 1, \ldots, M_i$. If i is an external stage, then $S = 0, 1, \ldots, M_i$.

The functional equation is evaluated for all possible integer incoming service times and feasible incoming cumulative costs for node i. That is, for $C \in XI_i$ and $SI = 0, 1, \ldots, M_i - \min\{T_{ij}\}$ for $0 \le j \le O_i$.

6.3.3 Dynamic programming algorithm

The dynamic programming algorithm is now as follows:

1. For i := 1 to N-1

    2. If p(i) is downstream of i, evaluate $f_i(C,S)$ for $S = 0, 1, \ldots M_i$ and $C \in X_i$.

    3. If p(i) is upstream of i, evaluate $F_i(SI,CI)$ for $SI = 0, 1, \ldots M_i - \min\{T_{ij}\}$ and $CI \in XI_i$.

4. For i := N evaluate $F_i(SI,CI)$ for $SI = 0, 1, \ldots M_i - \min\{T_{ij}\}$ and $CI \in XI_i$.

5. Minimize $F_N(SI,CI)$ for $SI = 0, 1, \ldots M_N - \min\{T_{Nj}\}$ and $CI \in XI_N$ to obtain the optimal objective function value.

This procedure finds the optimal objective function value; to find an optimal set of options and service times entails the standard backtracking procedure for a dynamic program.

To summarize, at each stage of the dynamic program, we find the minimum supply chain costs for the sub-network with node set N_i, as a function of the state variables. The state variable depends on the direction of the arc that connects the sub-network N_i to the rest of the network. When the connecting arc originates in N_i, then the state variable is the outbound service time (step 2); otherwise, the state variable is the inbound service time (step 3). We number the nodes so that the functions required to evaluate either $f_i(S,C)$ or $FI_i(SI,CI)$ have been determined prior to stage i in the dynamic program. At stage N (step 4), we determine the inventory costs for the entire network as a function of the inbound service time and incoming cumulative cost to node N. At step 5, we optimize over the states at node N to find the optimal inventory cost.

7 Example[5]

This section presents a real world example for the supply chain configuration problem. The section is structured as follows: In Section 7.1, we provide background on the sponsor company's current supply chain design process. In Section 7.2, we present a realistic example and the associated analysis. In Section 7.3, we present some general results and conclusions from this work.

7.1 Current process description

The company currently employs a target costing approach when designing new product supply chains. Ansari and Bell (1997) contains an extensive review of the state-of-the-art in target costing. In brief, the market price for the product is set outside of the product design group. Two common reasons for this are when the product faces many competitors, implying that the firm will be a price taker, and when another department within the company, for example marketing, specifies the product's selling price. Next, a gross margin for the product is specified, typically by senior management or corporate finance. The combination of the prespecified selling price and the gross margin target dictate the product's maximum unit cost.

The product design team uses the maximum unit cost as an upper bound on the product's unit manufacturing cost (UMC). UMC is defined as the sum of the direct costs associated with the production of a single unit of product. Typical costs include raw material costs, the processing cost at each stage, and transportation costs. The UMC acts as an overall

[5] The data in this section has been disguised to protect the company's proprietary information. The insights drawn from the disguised data are the same insights that were drawn from the real data. The data in this section

budget for the product, and this budget is then allocated to each of the product's subassemblies.

From an organizational perspective, the supply chain development core team is composed of an early supply chain enabler and one or two representatives associated with each of the product's major subassemblies. The early supply chain enabler is responsible for shepherding the product through the product development process. She is brought in during the early design phase and will stay with the project until it achieves volume production.

The core team will allocate the UMC across the major subassemblies. This is not an arbitrary process. The team will rely on competitive analysis, past product history, future cost estimates, and value engineering when making these decisions. Once the subassembly budgets are set, the design teams for each subassembly are charged with producing a subassembly that can provide the functionality required subject to the subassembly's budget constraint. Even if these groups incorporate multidisciplinary teams and concurrent engineering, the groups will still be operating within their own budget constraints.

In much the same way that the UMC is allocated to the subassemblies, each subassembly group must then determine what processes and components to use. There are numerous factors to consider when sourcing a component, some of which include functionality, price, vendor delivery history, vendor quality and vendor flexibility.

Since many of these factors are difficult to quantify, the team establishes a minimum threshold for each of the intangible factors. If a component exceeds each of the thresholds, then it can be considered. In the context of the supply chain configuration problem, an option

---

is based on a recently completed supply chain; hence, results described here were not implemented. We are currently in the process of applying the research to a supply chain that is in the design phase.

will be defined as a {cost added, production lead-time} pairing that satisfies all of the company's intangible factors.

The company's current practice can be described as choosing the component with the least unit cost among all of the components that can be considered. In the framework of the supply chain configuration problem, this corresponds to choosing the option with the least cost added at each stage, regardless of its production lead time. This practice minimizes the product's UMC. While this is admittedly a heuristic, there are several reasons why the company does this. First, as mentioned earlier, all of the other factors besides cost are difficult, if not impossible, to quantify. For example, the company only wants to do business with suppliers that have been certified. The certification process involves a rigorous review of the supplier's quality practices. But given two certified suppliers, there is no mechanism to view one supplier as superior to the other. Second, the UMC of the product will dictate whether or not the business case to launch the product is successful. If the UMC is not low enough to meet the gross margin target, then the project will be terminated. Therefore, there is tremendous pressure to meet the UMC target. Finally, the team that designs the supply chain is not the same team that has to manage the completed supply chain. Although choosing parts with long lead-times might increase significantly the supply chain's safety stock requirements, this dynamic has not been explicitly considered during the new product's business case analysis.

7.2 Digital Capture Device Example

The product that we studied can be described as a digital capture device. The product converts an analog input into a digital form. Both scanners and digital cameras satisfy this high-level description.

The product consists of three major subassemblies: the imager, the circuit board, and the base assembly. The imager captures the analog input. It is the subassembly that distinguishes the product in the marketplace. The imager is created in a four-stage process that begins as raw silicate and ends as a completed charge coupled device (CCD). The circuit board converts the analog input into a digital output. To create the circuit board, components are purchased from external vendors and assembled in-house. The base assembly has two components: the base and an accessory. Both components are purchased from an external vendor. The vendor must first modify the accessory before it can be delivered to the company.

The assembly process for the digital capture device involves fitting together the subassemblies and quality testing. Finally, the product supplies two different markets: US demand and export demand.

A graphical depiction of the supply chain is shown below:

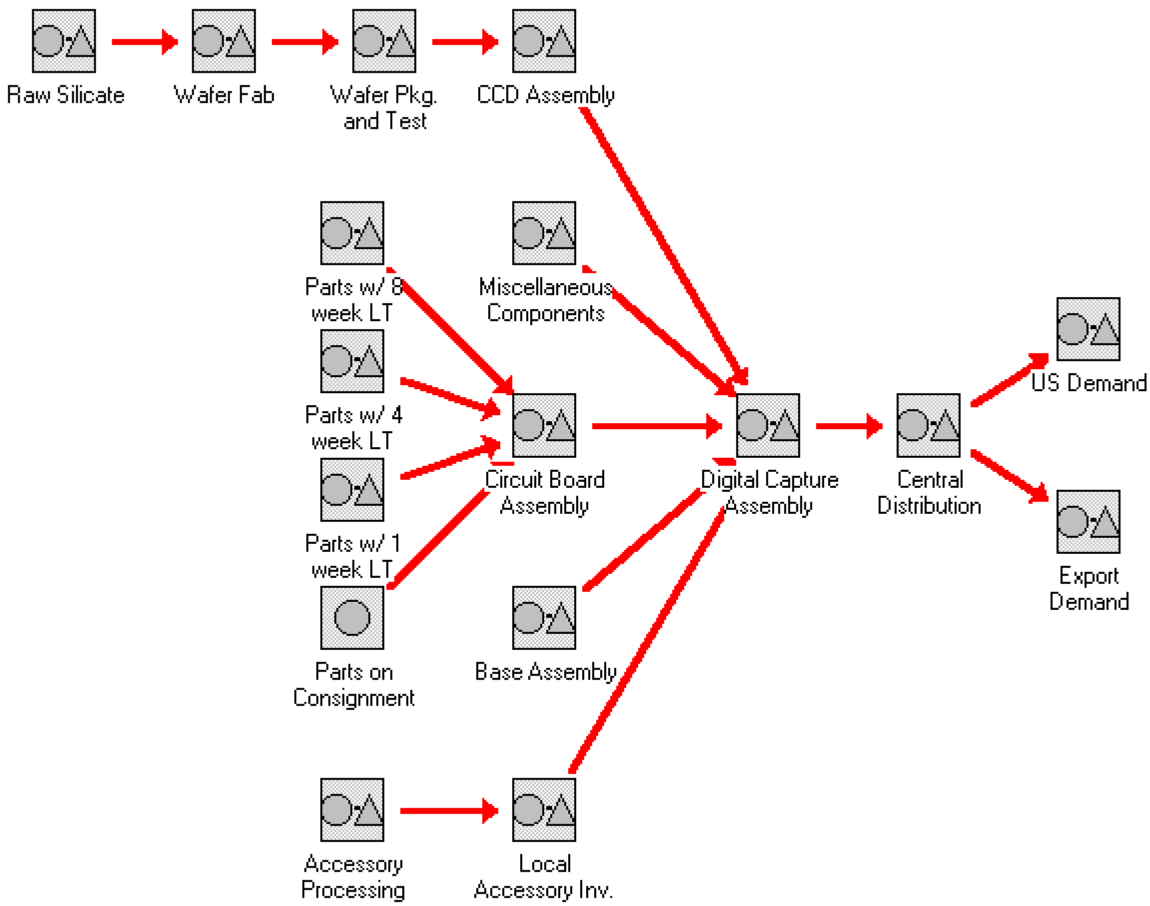

Figure 7-1: Digital Capture Supply Chain

The imager subassembly consists of the four stages at the top of the figure. Raw silicate is fabricated into imagers which are then packaged and tested. An imager is then mounted onto a stand to form the CCD. The components for the circuit board are grouped according to their traditional procurement lead times. The base assembly and accessory are depicted in accordance with their previous descriptions. After the digital capture product is assembled, it then goes through central distribution from where it satisfies either US or export demand.

The table below contains the options available when sourcing this supply chain.

Component/Process Description	Option	Production Time	Cost
Raw Silicate	1	60	$5.00
	2	20	$7.50
Wafer Fab	1	30	$800.00
	2	8	$825.00
Wafer Pkg. and Test	1	10	$200.00
	2	5	$225.00
CCD Assembly	1	5	$200.00
	2	2	$250.00
Miscellaneous Components	1	30	$200.00
Parts w/ 8 Week LT	1	40	$105.00
	2	20	$107.62
	3	10	$108.96
	4	0	$110.32
Parts w/ 4 Week LT	1	20	$175.00
	2	10	$177.18
	3	0	$179.39
Parts w/ 2 Week LT	1	10	$200.00
	2	0	$202.50
Parts on Consignment	1	0	$225.00
Circuit Board Assembly	1	20	$225.00
	2	5	$300.00
Base Assembly	1	70	$650.00
	2	30	$665.00
Accessory Processing	1	40	$100.00
Local Accessory Inv.	1	10	$60.00
Digital Capture Assembly	1	6	$420.00
	2	3	$520.00
Central Distribution	1	5	$180.00
US Demand	1	5	$12.00
	2	1	$25.00
Export Demand	1	11	$15.00
	2	2	$40.00

Table 7-1: Options for Digital Capture Product

The company operates on a five day work week and there are two hundred fifty days in the year. The annual holding cost rate is thirty percent. The company seeks to minimize the total supply chain configuration cost incurred over one year.

For each stage, option 1 reflects the option that was implemented for the existing supply chain. The additional options were judged by the materials management group to reflect {cost added, production leadtime} pairings that were alternatives to the options selected.

For the circuit board's raw materials, the different options refer to different classes of service that the vendor is willing to provide. The head of materials management for the electronics subassembly estimated that the cost of converting an eight week leadtime part to a consignment part would equal 5% of the part's eight week selling price. We used this information to estimate the cost of reducing one week of leadtime for each electronic part as 0.625% of the part's selling price.

As a rule of thumb, the company valued one hour of processing time at a stage at $50 per hour. Recall that the definition of production time includes the waiting time at a stage plus the actual processing time at the stage. Therefore, a slight increase in the processing time at a stage can dramatically reduce the stage's production time. For example, by adding $25 to the cost of wafer fab, the production time was reduced to eight days. A similar analysis was performed for wafer packaging and test, CCD assembly, and the assembly stages.

The two demand stages represent the delivery of product to the company's retail stores. The maximum service time for each of the demand stages equals zero. That is, they must provide immediate service to external customers. In the case of US demand, the product can either be shipped by ground transportation at a cost of twelve dollars and a transportation time of five days or it can be shipped by air at a cost of twenty five dollars with a one day transportation time. Export demand can be satisfied in a similar manner, albeit with different costs and transportation times.

The current product is an improved version of an existing product. Therefore, the company used the previous product's sales as well as market forecasts when determining the demand requirements for the supply chain. For US demand, the mean daily demand and standard deviation of demand were estimated as 15 and 9. For Export demand, the estimates were 4 and 2, respectively. At each of the demand stages, the demand bound was estimated as:

$$D_j(\tau) = \tau\mu + k\sigma\sqrt{\tau}$$

where τ is the net replenishment time, and μ and σ refer to the stage's mean and standard deviation of demand. The constant k was chosen to equal 1.645. The supply chain group felt that this demand bound captured the appropriate level of demand that they wanted to configure their system to meet using safety stock.

7.2.1 Minimizing UMC heuristic

The minimizing UMC heuristic consists of choosing the least cost option at each stage. For the example in Table 7-1, this corresponds to choosing option 1 for each function. A summary of the costs at the subassembly level is shown below:

Major Function	Cost	% of Total
Wafer	$1,205.00	31.95%
Base Platform	$810.00	21.47%
CBA	$930.00	24.66%
Misc	$200.00	5.30%
Assembly	$420.00	11.13%
Distribution	$207.00	5.49%
Total	$3,772.00	

Given that there is only one option at each stage, the only optimization to be done is the optimization of the safety stock levels across the supply chain; the expected pipeline stock cost and cost of goods sold are constant when there is only one option per stage. The optimal service times across the supply chain are shown in Figure 7-2. As a point of reference, the production times at each stage are shown in Figure 7-3.

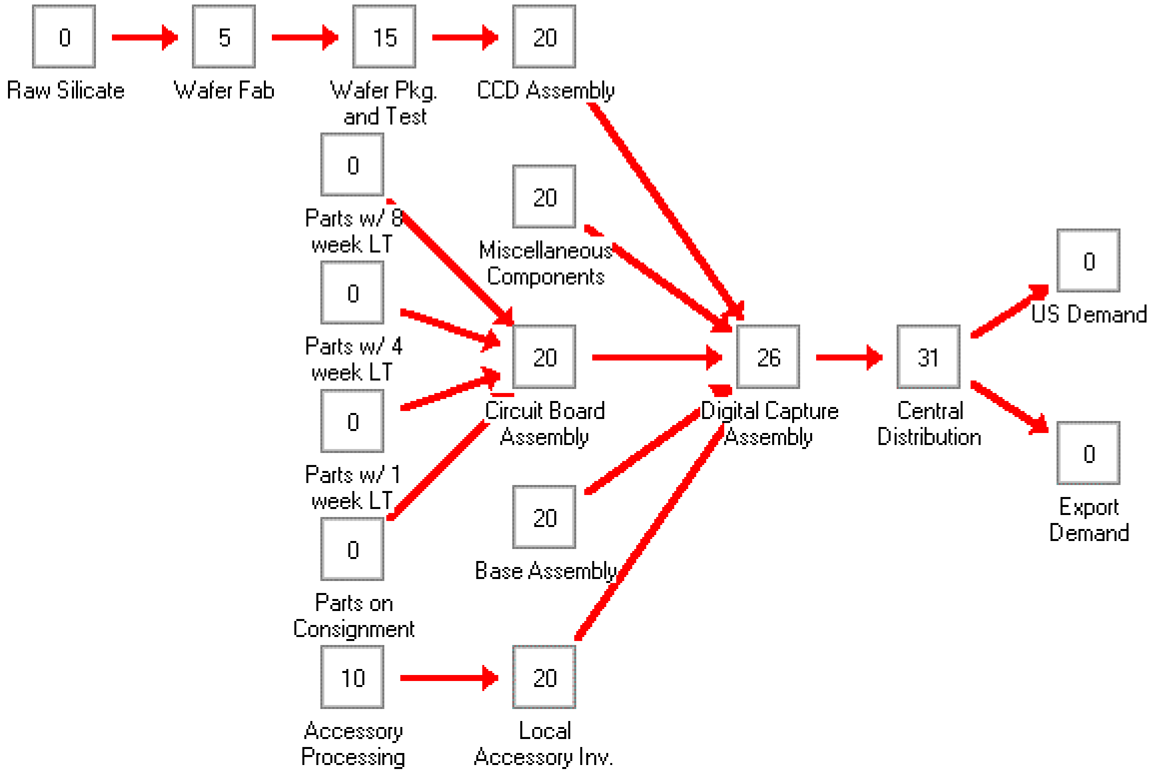

Figure 7-2: Optimal Service Times for Min UMC Heuristic

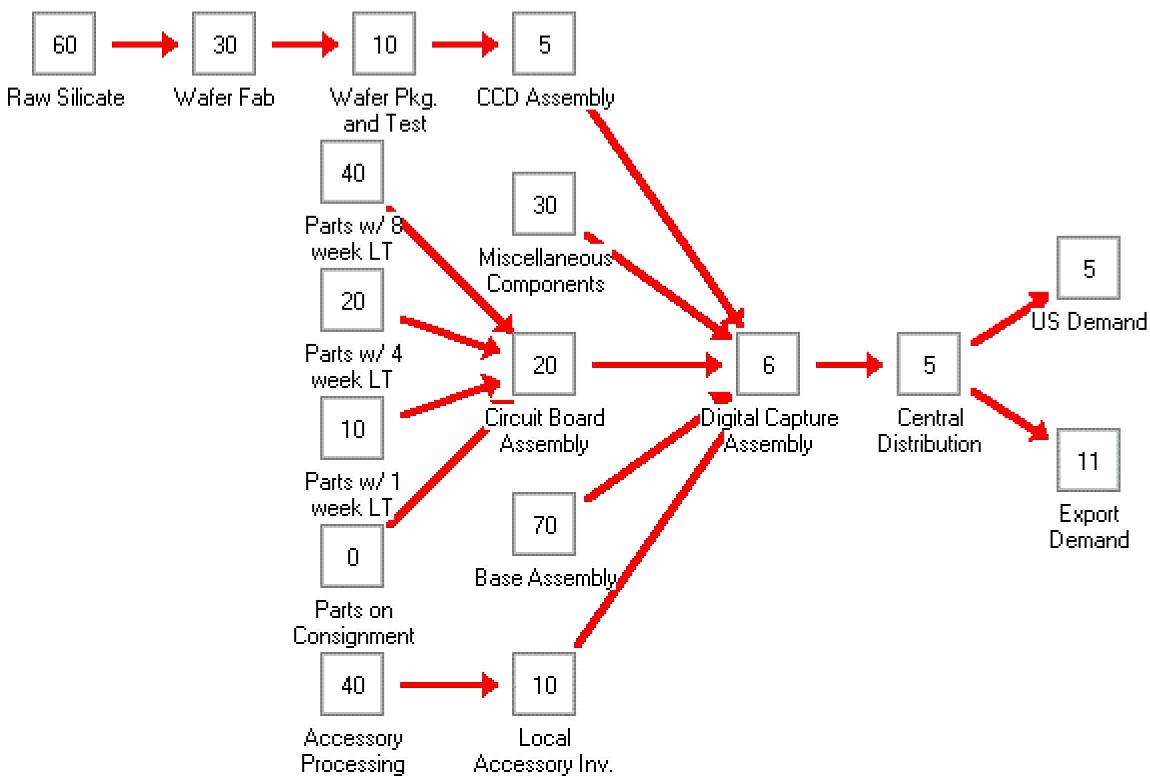

Figure 7-3: Production Times for Min UMC Heuristic

The optimal safety stock policy exhibits a dominant path. The dominant path is a serial line that starts at a raw material stage and stretches to the demand stages. Minimizing the service times along this path dictates the service times across the entire supply chain. Stages that are not on the dominant path set their service times as high as possible without changing the net replenishment times for any stages on the dominant path. In the Min UMC supply chain, the dominant path starts at the parts with a one week leadtime and ends at the distribution stages. Since the distribution stages are each linked to the Central Distribution stage, the two external stages share the same dominant path.

The optimal safety stock policy is to position several decoupling safety stocks across the supply chain. Figure 7-4 provides a graphical representation of the supply chain's optimal safety stock policy.

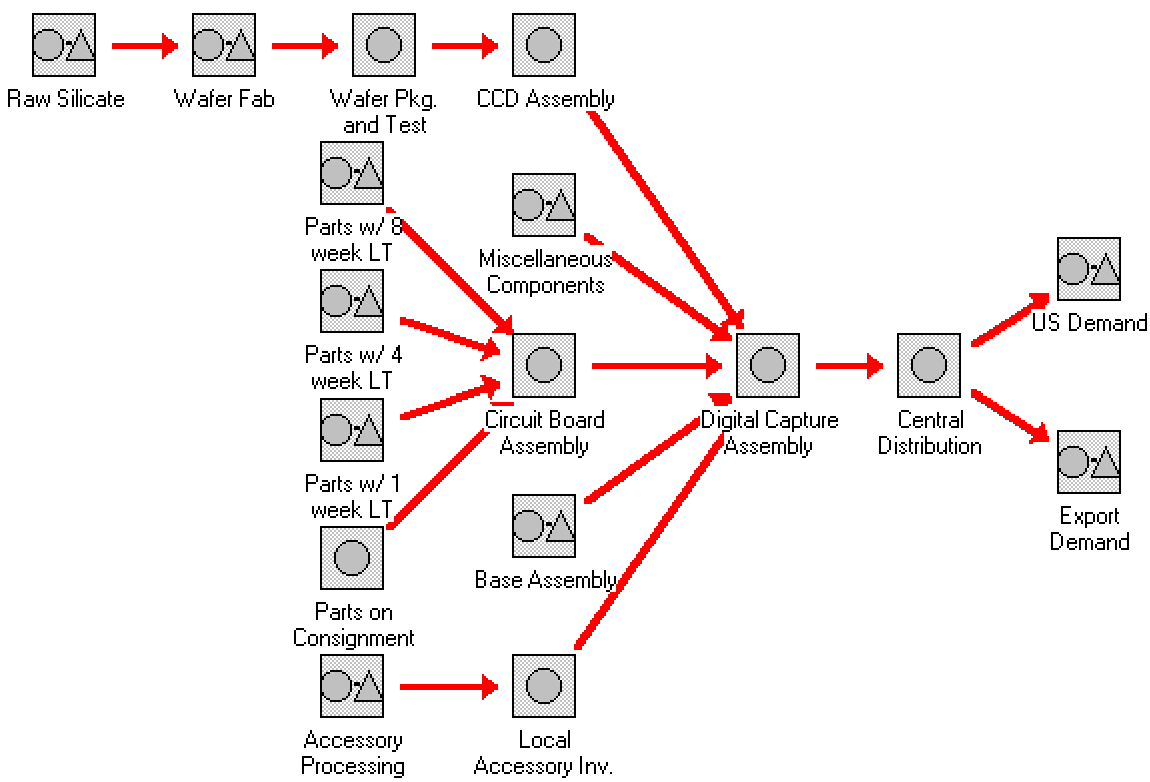

Figure 7-4: Optimal Safety Stock Placement for Min UMC Heuristic

In the figure, a circle denotes a processing operation and a triangle denotes a safety stock location. Safety stock is held at both of the demand stages. Since both of these stages must quote a service time of zero, they have to stock inventory. The demand stages are both quoted a service time of 31 days. Given these service times, none of the subassemblies have to hold safety stock in a completed form. In fact, the safety stock policies of the subassemblies can best be described as policies that minimize their individual portions of the supply chain given that they can each quote an outgoing service time of 20 days (recall that if they quoted more than 20 days, the net replenishment time of downstream stages like Digital Capture Assembly would have to change). For the circuit board, this translates into storing a safety stock for each of the raw material stages. For the imager supply chain, the optimal solution is to hold raw silicate and have wafer fab quote a service time of 5. Finally, stages like the accessory and the base assembly quote service times of 20, holding their inventory as far upstream as possible.

A summary of the configuration's costs are shown below:

SS Cost	$178,386
PS Cost	$979,127
COGS	$17,848,750
Total	$19,006,263

Table 7-2: Cost Summary for Min UMC Heuristic

The safety stock and pipeline stock costs reflect the company's 30% carrying cost. Therefore, the initial investment in safety stock and pipeline stock to create the supply chain equals $3,858,376 (this can be seen by dividing the pipeline and safety stock costs by 0.3). The expected demand over the course of one year is 4,750 units; this is found by multiplying the expected daily demand (19) by the number of days in the year (250). Since a completed unit costs either $3,757 or $3,760, depending on the customer region, COGS dominates the total supply chain configuration cost.

7.2.2 Minimizing production time heuristic

The minimizing production time heuristic chooses the option at each stage with the least production time. This corresponds to choosing the option with the highest index for each function in Table 7-1. A summary of the costs at the subassembly level is shown below:

Major Function	Cost	% of Total
Wafer	$1,307.50	31.78%
Base Platform	$825.00	20.05%
CBA	$1,017.21	24.72%
Misc	$200.00	4.86%
Assembly	$520.00	12.64%
Distribution	$245.00	5.95%
Total	$4,114.71	

Like the minimum UMC heuristic, the minimum production time heuristic leaves only one option at each stage. The optimal service times across the supply chain are shown in Figure 7-5. As a point of reference, the production times at each stage are shown in Figure 7-6.

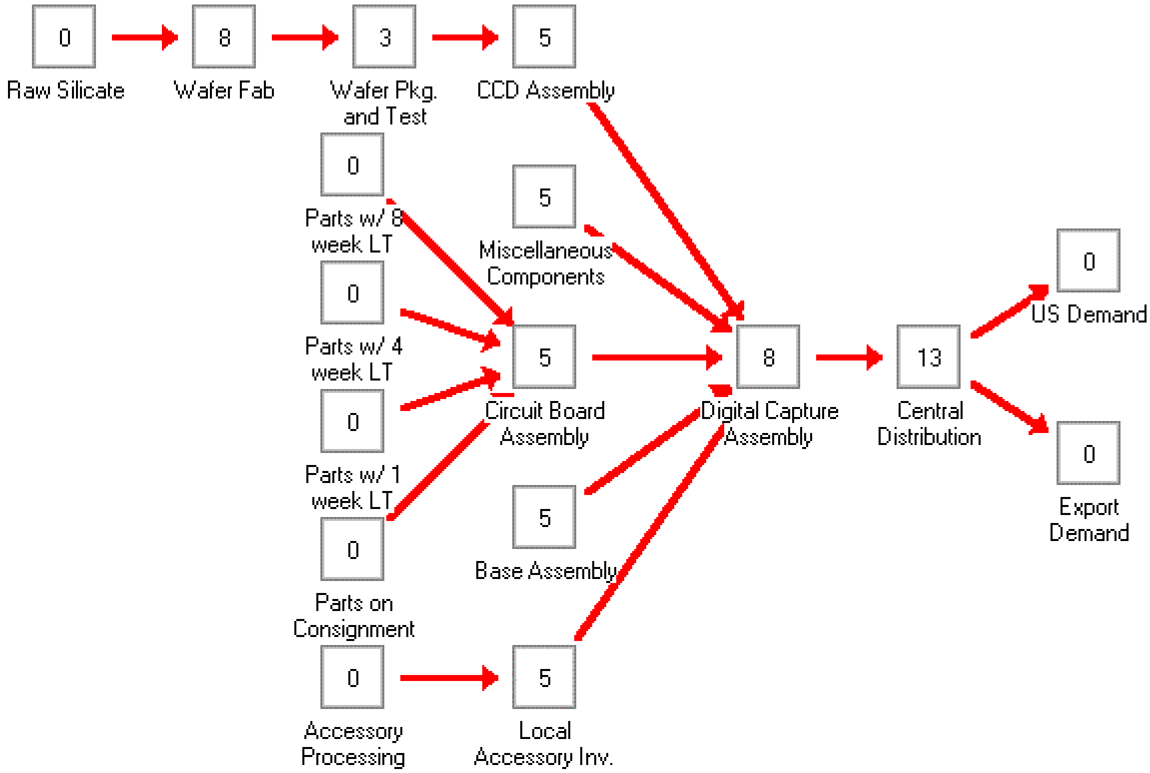

Figure 7-5: Optimal Service Times for Min Production Time Heuristic

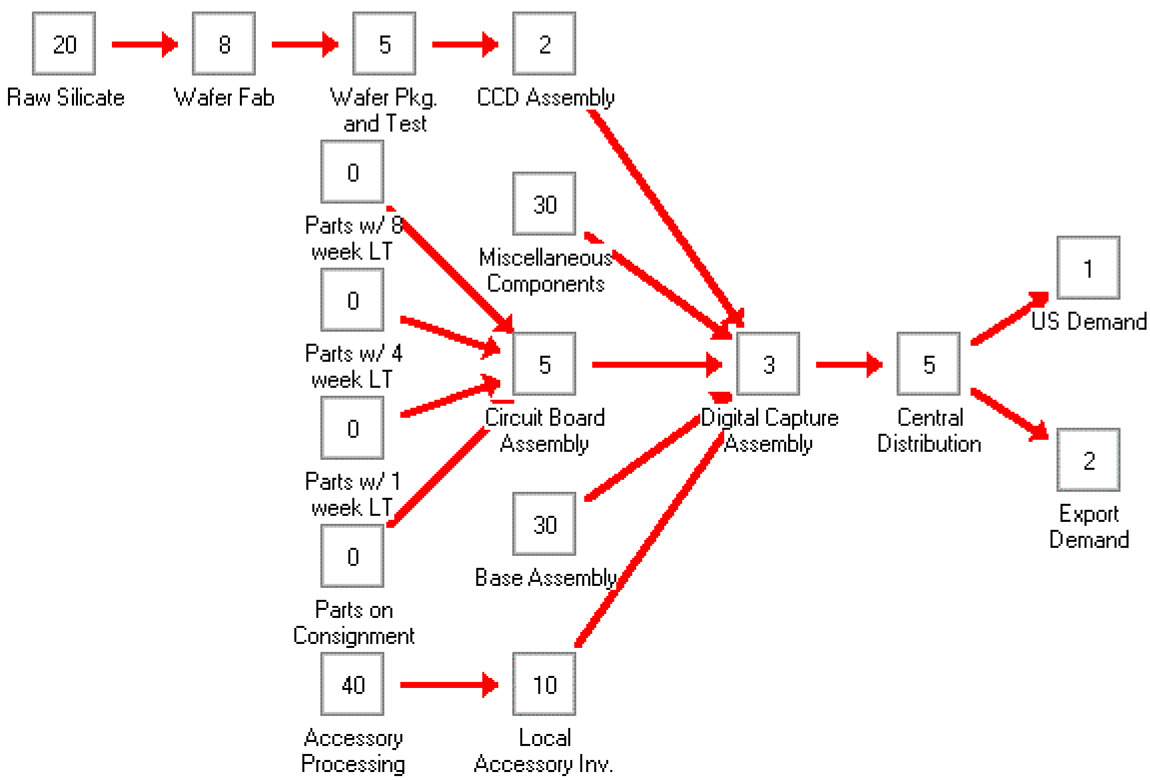

Figure 7-6: Production Times for Min Production Time Heuristic

In the Min production time configuration, the dominant path originates at the electronics components and ends at the demand nodes. The optimal stocking policy is represented graphically below:

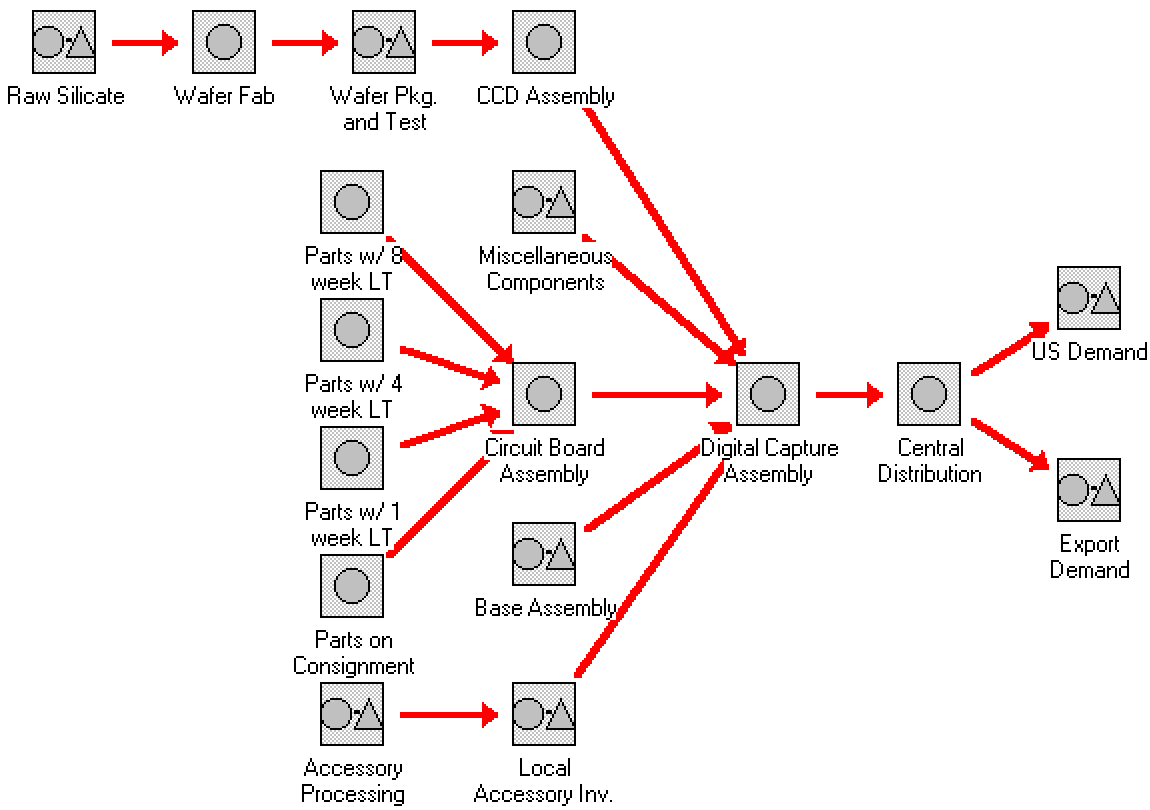

Figure 7-7: Optimal Safety Stock Placement for Min Production Time Heuristic

An intuitive explanation for the optimal stocking policy is that the minimum production time heuristic causes all of the electronic parts to be held on consignment. Therefore, the compressed leadtime of the entire circuit board assembly makes it attractive to hold as little inventory as possible between the circuit board and the final product; recall that the demand stages must stock inventory due to their service time commitments. The other subassemblies then locally optimize their own portion of the supply chain subject to the constraint that they must quote a service time of 5 or less to Digital Capture Assembly. Since each subassembly's maximum replenishment time is more than 5 days, they will each quote Digital Capture

Assembly exactly 5 days. The imager supply chain is the only subassembly that requires optimization; for all the other stages it is either stock or don't stock. The high cost of the Wafer Fabrication makes it optimal to hold as little inventory as possible after the fabrication step.

A summary of the configuration's costs is shown below:

SS Cost	$122,890
PS Cost	$465,886
COGS	$19,369,873
Total	$19,958,648

Table 7-3: Cost Summary for Min Production Time Heuristic

The safety stock and pipeline stock costs are dramatically reduced due to the shortened production times across the network. However, this comes at a significant cost since the product's UMC increases by nine percent. The initial investment in safety stock and pipeline stock to create the supply chain equals $1,962,586 (this can be seen by dividing the pipeline and safety stock costs by 0.3). The minimum production time heuristic results in a supply chain configuration cost that exceeds the minimum UMC heuristic by $950,000.

7.2.3 Supply chain configuration optimization

We will now apply the algorithm presented in Section 6 to the example. The following options are selected at each stage:

Component/Process Description	Option	Production Time	Cost
Raw Silicate	1	60	$ 5.00
Wafer Fab	1	30	$ 800.00
Wafer Pkg. and Test	1	10	$ 200.00
CCD Assembly	1	5	$ 200.00
Miscellaneous Components	1	30	$ 200.00
Parts w/ 8 Week LT	3	10	$ 108.96
Parts w/ 4 Week LT	2	10	$ 177.18
Parts w/ 2 Week LT	1	10	$ 200.00
Parts on Consignment	1	0	$ 225.00
Circuit Board Assembly	1	20	$ 225.00
Base Assembly	2	30	$ 665.00
Accessory Processing	1	40	$ 100.00
Local Accessory Inv.	1	10	$ 60.00
Digital Capture Device Assembly	1	6	$ 420.00
Central Distribution	1	5	$ 180.00
US Demand	2	1	$ 25.00
Export Demand	2	2	$ 40.00

Table 7-4: Options Selected Using Optimization Algorithm

In this configuration, the electronic components not held on consignment have a common two week procurement lead-time. Also, the base assembly's lead-time has been shortened to thirty days and the air shipment of finished goods is preferred over the longer ground shipment option.

A summary of the costs at the subassembly level is shown below:

Major Function	Cost	% of Total
Wafer	$1,205.00	31.45%
Base Platform	$825.00	21.53%
Circuit Board	$936.14	24.44%
Misc	$200.00	5.22%
Assembly	$420.00	10.96%
Distribution	$245.00	6.39%
Total	$3,831.14	

The optimal service times across the supply chain are shown in Figure 7-8. As a point of reference, the production times at each stage are shown in Figure 7-9.

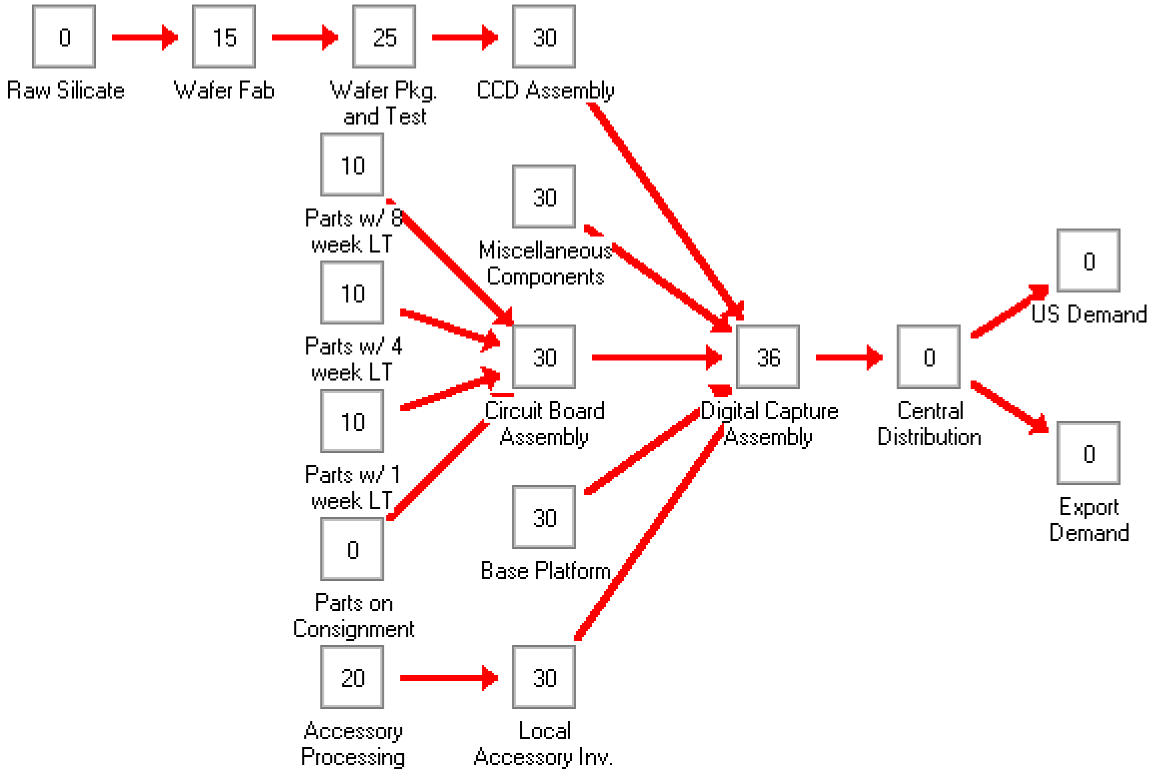

Figure 7-8: Optimal Service Times for Optimization Algorithm

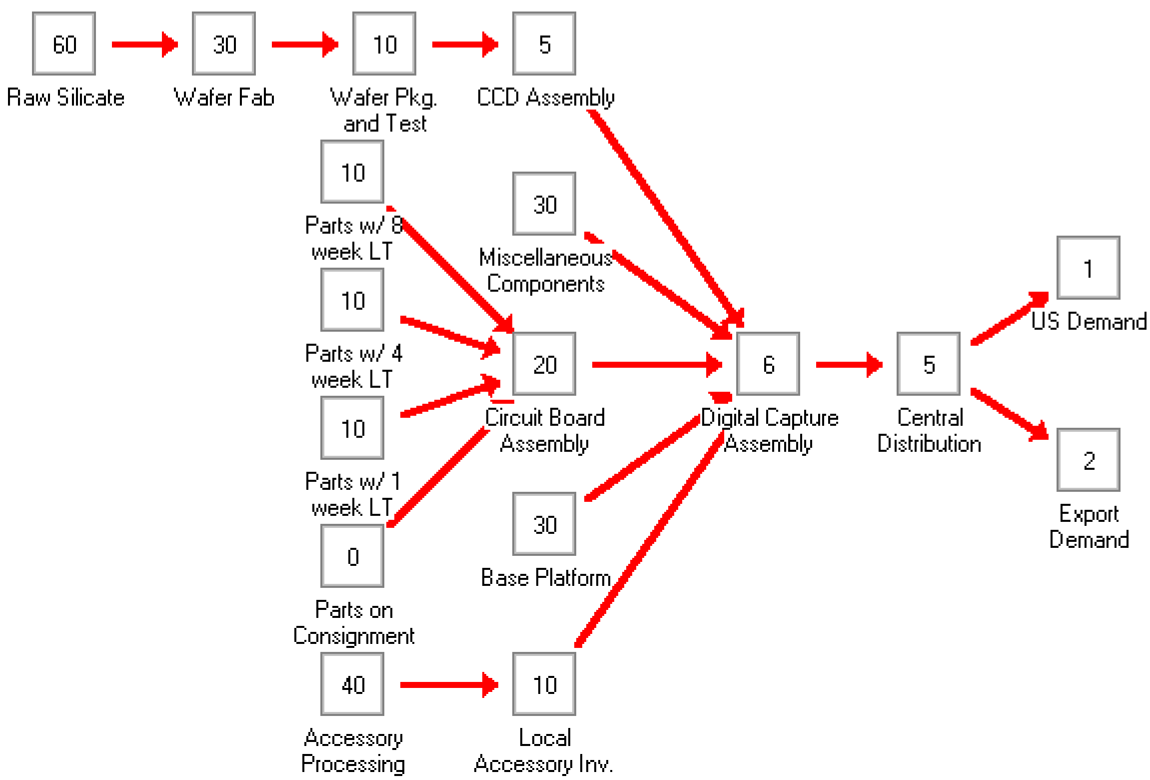

Figure 7-9: Production Times for Optimization Algorithm

The dominant path still originates at the electronics components and ends at the demand nodes. The optimal stocking policy is represented graphically below:

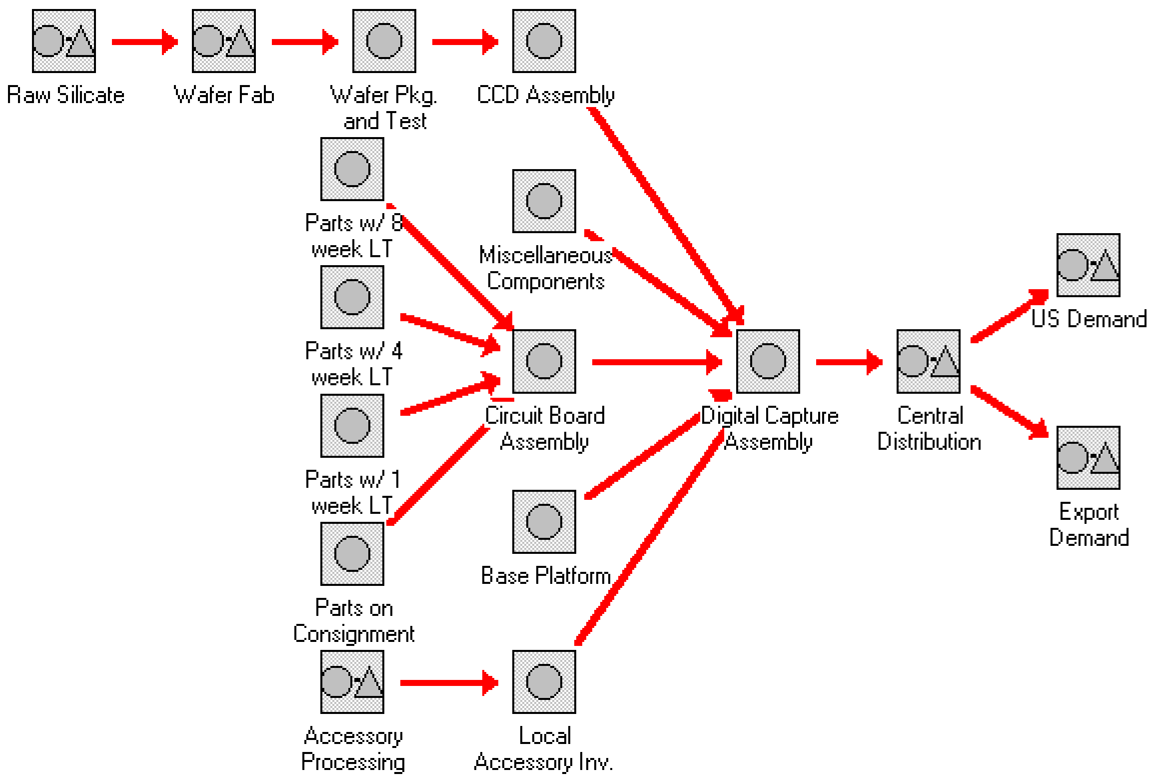

Figure 7-10: Optimal Safety Stock Placement for Optimization Algorithm

The optimal policy holds a decoupling inventory at the central distribution center. By holding inventory at the Central Distribution Center, each of the external stages can hold significantly less inventory (since the incoming service time to each of the demand stages equals zero, their net replenishment time equals their production time). The primary reason this is optimal is because the higher shipment costs make it less attractive to hold inventory at the external stages. By choosing a one week production time for each of the electronics components (besides the consignment stage, which is restricted to a zero production time option) and choosing the least production time base platform option, the optimal solution is one where the upstream assemblies are "balanced." That is, each subassembly is configured in the optimal way to quote a service time of 30 to the Digital Capture Assembly.

A summary of the configuration's costs is shown below:

SS Cost	$148,254
PS Cost	$700,097
COGS	$18,022,915
Total	$18,871,266

Table 7-5: Cost Summary for Optimization Algorithm

The initial investment in safety stock and pipeline stock to create the supply chain equals $2,827,837. This configuration increases the UMC by 1.6% over the min UMC heuristic but decreases the total configuration cost by $135,000. This represents a per unit savings of $28.42.

To help put this cost savings into perspective, the following chart summarizes the costs for the Min UMC configuration when each stage holds safety stock (this situation is depicted in Figure 7-1):

SS Cost	$237,678
PS Cost	$979,127
COGS	$17,848,750
Total	$19,065,555

Table 7-6: Cost Summary for Min UMC Heuristic with Service Times Equal to Zero

The Min UMC heuristic with service times equal to zero is the most accurate representation of the company's implemented supply chain. The savings generated by optimizing the safety stock levels without changing the supply chain's configuration equals $59,292 (this optimized case is actually the Min UMC heuristic presented in Section 7.2.1). The savings generated by jointly optimizing the safety stock levels and the supply chain's configuration total $194,289. Therefore, jointly optimizing both the configuration and the safety stock placement will save three times as much as leaving the configuration unchanged and only optimizing the safety stock placement.

Also, it is important to note that implementing the optimal policy is an extremely easy matter. The difficult step in the supply chain design process is the identification of the parts that exceed all of the intangible requirements. However, this step must be done regardless of which option is eventually chosen. The optimization algorithm just optimally picks among the set of options that are all sufficient to satisfy the product's needs.

Finally, it is interesting to note that although the overall UMC has not increased by much, there is no way the design team would have known to pick this configuration. Table 7-7 summarizes the costs at the subassembly level for the Min UMC heuristic and the Optimization Algorithm's configuration.

Major Function	Subassembly UMC under Min UMC Heuristic	Subassembly UMC under Optimal Configuration	% difference
Wafer	$1,205	$1,205	0.00%
Base Platform	$810	$825	1.85%
Circuit Board	$930	$936	0.66%
Misc	$200	$200	0.00%
Assembly	$420	$420	0.00%
Distribution	$207	$245	18.36%
Total	$3,772	$3,831	1.54%

Table 7-7: Comparison of Option Costs for Min UMC and Optimal Configuration

For subassemblies like the base platform, increasing the UMC by $15 is a dramatic increase that would not be authorized without the kind of analysis presented in this section. The same is true of the adoption of premium freight.

The optimization algorithm also neglected to make some choices that the team might have considered "obvious" choices. For example, the higher cost raw silicate option was not selected. Conventional wisdom might have led one to believe that this option would be selected due to the fact that the imager subassembly is an expensive component with a long maximum replenishment time. And with a modest increase in the subassembly's cost, the maximum replenishment time could be significantly shortened. However, the decrease in production time did not offset the subsequent increase in the cost.

7.3 General Conclusions

Based on the current analysis that has been performed, some general hypotheses can be formulated. First, the farther upstream the supply chain, the less likely it will be optimal to choose a significantly higher UMC option. The reason is that choosing the higher cost option not only increases the product's UMC, it also increases the pipeline and safety stock cost at downstream stages. Furthermore, since these raw materials are in their cheapest state, it is less costly to just hold a safety stock of raw components, thereby decoupling them from the rest of the supply chain. Therefore, when choosing a higher cost option that is upstream, the savings will have to be truly dramatic to justify the higher UMC.

Second, the larger the number of echelons in the supply chain, the larger the potential significance of this approach. More echelons imply more flexibility in setting up the supply chain. This gives the reduced safety stock and pipeline stock costs a greater opportunity to outweigh the increase in COGS. This insight should be tempered by the realization that any non-value added steps should be removed wherever possible. Before using the model, a general recommendation would be to remove any non-value added stages from the process.

Finally, given the fact that product life cycles are only getting shorter, it is generally a bad idea to make large investments in creating a supply chain. By making a larger dollar investment in safety stock and pipeline stock, it can be more difficult to effectively manage the "ramp down" phase of the product's life. In the "ramp down" phase, safety stock and pipeline stock are drawn out of the system and used to fulfill demand. The goal is to leave as little inventory as possible in the supply chain when the product is terminated; if the effort is completely successful, no inventory will remain.

Since the Min UMC heuristic typically chooses the configuration with the longest lead-times, it requires significant pipeline and safety stock levels. This is the exact opposite situation that a manager would want to create. However, the desire to create a lower investment supply chain must be balanced against the dramatic increase in cost created by implementing the most responsive supply chain (which can be found by using the Min Production Time heuristic).

The objective of the supply chain configuration problem is to balance these two competing interests. An effective way for managers to strike this balance is through the setting of the holding cost rate. The holding cost rate can be used to help gauge how much risk the company associates with making a large investment in safety and pipeline stock. If the company places a significant cost on making large initial investments in the supply chain, they can attach a higher holding cost rate. This would make the safety stock cost and pipeline stock cost a much higher total proportion of the configuration cost, thereby acting to mitigate the increase in COGS due to choosing higher cost, lower leadtime options. Conversely, if the company is not concerned about making a large initial investment, they can choose a lower holding cost rate.

To demonstrate this effect, the following table summarizes the optimal costs for each of the three configuration approaches when the holding cost rate is 15%, 30%, 45% and 60%.

	Minimum UMC Heuristic	Minimum Production Time Heuristic	Supply Chain Configuration Algorithm
Holding Cost - 15%			
Safety Stock Cost	$89,193	$61,445	$81,006
Pipeline Stock Cost	$489,563	$232,943	$417,484
COGS	$17,848,750	$19,369,873	$17,920,000
Total Configuration Cost	$18,427,506	$19,664,260	$18,418,489
Investment Cost	$3,858,375	$1,962,584	$3,323,262
Length of Longest Path	127 days	45 days	127 days
Holding Cost - 30%			
Safety Stock Cost	$178,386	$122,890	$148,254
Pipeline Stock Cost	$979,127	$465,886	$700,097
COGS	$17,848,750	$19,369,873	$18,022,915
Total Configuration Cost	$19,006,263	$19,958,648	$18,871,266
Investment Cost	$3,858,375	$1,962,584	$2,827,837
Length of Longest Path	127 days	45 days	118 days
Holding Cost - 45%			
Safety Stock Cost	$267,579	$184,334	$222,699
Pipeline Stock Cost	$1,468,690	$698,828	$1,010,252
COGS	$17,848,750	$19,369,873	$18,051,748
Total Configuration Cost	$19,585,019	$20,253,035	$19,284,699
Investment Cost	$3,858,375	$1,962,584	$2,739,892
Length of Longest Path	127 days	45 days	118 days
Holding Cost - 60%			
Safety Stock Cost	$356,772	$245,779	$270,291
Pipeline Stock Cost	$1,958,253	$931,771	$1,254,324
COGS	$17,848,750	$19,369,873	$18,170,498
Total Configuration Cost	$20,163,775	$20,547,423	$19,695,113
Investment Cost	$3,858,375	$1,962,584	$2,541,026
Length of Longest Path	127 days	45 days	96 days

Table 7-8: Configuration Cost Summary Table Under Different Holding Cost Rates

As the holding cost rate increases, the supply chain configuration algorithm chooses more higher cost, lower production leadtime options. This is demonstrated in the following table.

	15%	30%	45%	60%
Raw Silicate	1	1	1	1
Wafer Fab	1	1	1	2
Wafer Pkg. and Test	1	1	1	1
CCD Assembly	1	1	1	1
Miscellaneous Components	1	1	1	1
Parts w/ 8 Week LT	1	3	4	4
Parts w/ 4 Week LT	1	2	3	3
Parts w/ 2 Week LT	1	1	2	2
Parts on Consignment	1	1	1	1
Circuit Board Assembly	1	1	1	1
Base Assembly	2	2	2	2
Accessory Processing	1	1	1	1
Local Accessory Inv.	1	1	1	1
Digital Capture Device Assembly	1	1	1	1
Central Distribution	1	1	1	1
US Demand	1	2	2	2
Export Demand	1	2	2	2

Table 7-9: Optimal Supply Chain Configuration Under Different Holding Cost Rates

When the holding cost rate is low, COGS dominates the total configuration cost. Therefore, the minimum cost heuristic produces a solution that is very close to the optimal solution. But as the holding cost rate increases, the supply chain configuration algorithm creates a supply chain that comes closer to creating the supply chain created using the minimum production time heuristic.

8 Appendix I – Labeling Procedure[6]

8.1 Algorithm for Spanning Tree

We describe in this section the labeling procedure when the underlying network for the supply chain is a spanning tree, like in the figure below.

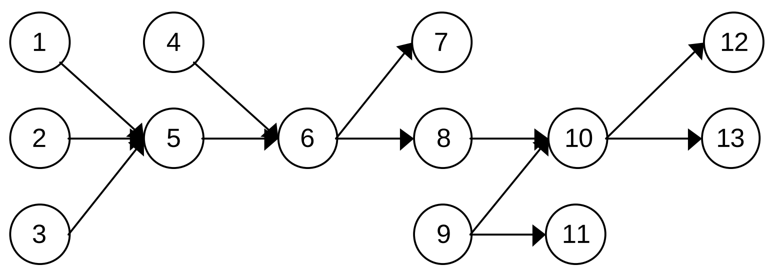

Figure 1: Spanning Tree

For a spanning tree, there is not a readily apparent ordering of the nodes by which the algorithm would proceed; indeed, we desire to sequence or number the nodes so that the algorithm is most efficient. The algorithm for labeling or re-numbering the nodes is as follows:

1. Start with all nodes in the unlabeled set, U.
2. Set k := 1
    3. Find a node $i \in U$ such that node i is adjacent to at most one other node in U. That is, the degree of node i is 0 or 1 in the sub-graph with node set U and arc set A defined on U.

[6] The labeling section is taken from Graves and Willems (1998).

4. Remove node i from set U and insert into the labeled set L; label node i with index k.
5. Stop if U is empty; otherwise set k:= k+1 and repeat steps 3 – 4.

For a spanning tree, it is easy to show that there will always be an unlabeled node in step 3 that is adjacent to at most one other unlabeled node. As a consequence, the algorithm will eventually label all of the nodes in N iterations. Indeed, we can show that each node labeled in the first N-1 steps is adjacent to exactly one other node in set U. That is, the nodes with labels 1, 2, … N-1 each have one adjacent node with a higher label; we define p(k) to be the node with higher label that is adjacent to node k, for k = 1, 2, … N-1. The node with label N obviously has no adjacent nodes with larger labels.

We assume in the following that the nodes in the spanning tree have been re-numbered according to this algorithm. For instance, we have used the algorithm to re-number the nodes in Figure 1 to produce Figure 2. Note that the labeling is not unique as there may be multiple choices for node i in step 3.

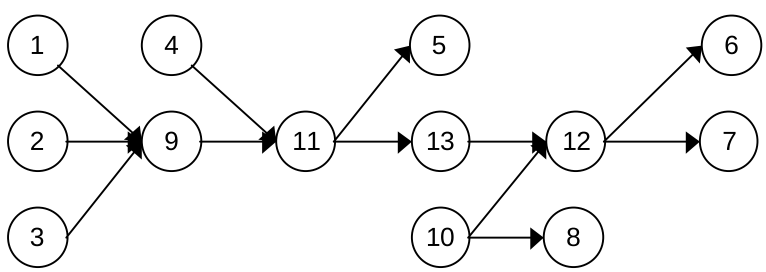

Figure 8-1: Renumbered Spanning Tree

For each node k we define N_k to be the subset of nodes {1, 2, ... k} that are connected to k on the sub-graph consisting of nodes {1, 2, … k}. We will use N_k to explain the dynamic programming recursion. We can determine N_k by the following equation:

$$N_k = \{k\} + \bigcup_{i<k,(i,k)\in A} N_i + \bigcup_{j<k,(k,j)\in A} N_j .$$

For instance, in Figure 2 N_k is {3} for k=3, {1, 2, 3, 9} for k=9, {1, 2, 3, 4, 5, 9, 11} for k=11 and {6, 7, 8, 10, 12} for k=12. We can compute N_k as part of the algorithm for re-numbering the nodes.

Part II: Part Selection in Multigeneration Products

9 Introduction

This section will look at determining optimal part selection strategies in multigeneration products. The goal is to develop a decision support tool that a product development team can use when they are determining what parts to design into their product. The model framework considers the development cost associated with redesigning the product, the part manufacturing cost and the level of functionality that the part must provide. The part selection problem seeks to choose the optimal set of parts that minimize the sum of the development and manufacturing costs subject to satisfying each period's functionality requirements.

The paper is structured as follows. In Section 10, the modeling framework is introduced. In Section 11, the different cases are formulated and solved. In Section 12, a brief numerical example is presented.

10 Model Introduction

This section presents the basics of the modeling framework. The decision variables and inputs are defined.

10.1 Performance definitions

10.1.1 Part performance level

A single attribute, or a collection of attributes, determines the performance level of each part. A mapping function transforms the part's attribute into a performance level that can range from negative to positive infinity. The nature of the attribute will dictate whether a part with a higher or lower performance level corresponds to a superior part. We assume that each part can be mapped into the performance level.

For some attributes, the mapping function is simply the multiplication of the technical attribute by a constant to create a unit-less index value. For example, for microprocessors, if the attribute of interest is speed, multiplying by 1/mhz creates a valid performance range; i.e., a 100 mhz processor has a performance of 100. For other attributes, like weight or picture quality, more complicated mapping functions are necessary.

For each part i, we let a_i denote its attribute of interest. The function $p_j(a_i)$ then converts part i's attribute into its corresponding performance level in period j. Note that the performance mapping function, $p_j()$, is period dependent. Thus, we can allow the part to decline relatively to other available parts. In addition, if the performance level is a function of multiple attributes, then a_i would be a vector of these attributes.

10.1.2 Performance requirement

Each period, there is a performance value that represents the ideal part performance level. Informally, this corresponds to the market's "sweet spot." If cost was not a concern and the set of available parts was sufficient to cover all possible performance levels, then each period the selected part's performance level would equal the period's performance requirement. However, since these two conditions are not met each period, it will not always be feasible, let alone optimal, to meet exactly the performance requirement each period.

The nature of performance requirement dictates the solution procedure employed. In Section 11, we will consider two cases: when the performance requirement is deterministic and when the performance requirement is an independent random variable. When the performance requirement is deterministic, each period's performance requirement is known with certainty at the start of the problem horizon. In the case where the performance requirement is a random variable, the probability distribution for each period's requirement is known at the start of the horizon.

10.2 Timing of Events

We consider a firm that is determining its part selection strategy for the next N periods, where time proceeds from period 1 to period N. At the start of each period, the period's performance requirement is realized. The firm then chooses the part that will be used to satisfy the current period's performance and demand requirements. After the part has been selected, the period's costs are incurred.

10.3 Part indexing

We let n denote the total number of distinct parts that are available in at least one of the N periods. We assume that the parts are indexed from 1 to n and that a part's index stays constant across periods. That is, if part i is available in two different periods, then part i refers to the same part. A further impact of this assumption is that the set of parts available each period will likely not be numbered contiguously. Since a part's performance level can be period dependent, we can not order the parts in an ascending or descending order based on their performance levels.

We let S_j denote the set of parts that are available in period j.

10.4 Costs

There are three relevant costs to consider: development cost, manufacturing cost and recycling cost.

10.4.1 Development cost

A development cost of K is incurred whenever the part used in the current period in the product differs from the part used in the previous period. Typical activities that must occur when a new part is selected include redesigning the interface between the part and the rest of the product, prototyping the new part, and certifying the part's supplier. Although the development cost could be both part and time dependent, we assume that it is a constant in our formulation. This is due to the example that motivated this research.

10.4.2 Manufacturing cost

Manufacturing cost is composed of the costs required to make a part in the current period. Typical costs include the procurement of raw materials and the transformation of the raw materials into completed parts. The period's unit manufacturing cost will depend on its initial cost net any discounts that can depend either on cumulative production volume or the length of time the part has been used. We denote the initial unit cost of part i in period j by c_{ij}.

This model considers both time and quantity discounts. In practice, both discounts are specified in the part's contract; this contract would be written in period t after the part is selected.

The time discount is a negotiated price break that occurs in every period the part is produced. For example, the discount might be 5% of the initial part cost for each period that the part is used. The volume discount is based on the part's cumulative production volume prior to the start of the current period. For example, the supplier might give a 1% discount for every 100,000 units purchased in earlier periods.

We let $e_{ij}(t,v)$ denote the discount rate for part i in period j given that the part was introduced in period t and the cumulative production up to period j equals v. It is defined below:

$$e_{ij}(t,v) = (j-t)\alpha_i + \left\lfloor \frac{v}{\nu_i} \right\rfloor \beta_i \qquad \text{for } j \geq t \tag{27}$$

where α_i is the time-dependent discount, β_i is the volume-dependent discount, and ν_i is the volume discount step size; these constants are all part specific.

10.4.3 Recycling cost

There are two ways to satisfy demand in the current period. First, as described in Section 10.4.2, they can produce new parts. Second, the firm can recycle parts that are returned (the company has an active and successful recycling program). The cost to recycle an existing part is significantly less than the cost of producing a new part. Recycling an existing part requires extricating the part from the housing and testing it to make sure that the part is still functional.

It is important to note that only the products containing the current part are worth recycling. Since the product had to be modified to accommodate the current part, older parts are unusable even if they still function properly. We let y_i denote the cost of remanufacturing part i. Remanufacturing is a labor-dominated process and as such does not depend on the period or the part's original manufacturing cost.

10.5 Production and demand requirements

Before we can formulate the part selection problem, we need to characterize the demand and recycling processes.

10.5.1 Demand process characterization

We assume that demand each period is deterministic. Demand in period j is denoted d_j.

10.5.2 Recycling process characterization

We are interested in characterizing the stream of recycled parts that are available to use in each period. In the context of recycling, let τ denote the useful life of the product. That is, a product produced in period t can only be recycled during the interval $[t+1, t+\tau]$. γ_k is a scalar that represents the fraction of parts used in period $t - k$ that are returned in period t. Since we can't receive more units than we shipped in a period, it must be true that:

$$\sum_{k=1}^{\tau} \gamma_k \le 1$$

where the constraint is satisfied with equality only when all the parts produced in a period are eventually returned within τ periods.

Let $q_j(t)$ denote the number of recycled parts that are available to satisfy demand in period j given that the part was introduced in period t. $q_j(t)$ is a function of the amount of the current part in circulation, and is computed as follows:

$$q_j(t) = \sum_{k=1}^{\ell} \gamma_k d_{j-k} \qquad \text{where } \ell = \min(\tau, t-j) \tag{28}$$

For example, if $\tau = 3$, the current period is 5 and the part was introduced in period 1, equation (28) would look like:

$$q_5(1) = \gamma_1 d_4 + \gamma_2 d_3 + \gamma_3 d_2.$$

10.5.3 Cumulative production recursion

The presence of recycling makes the calculation of the cumulative production slightly more cumbersome. The cumulative production is not simply the sum of the demands from previous periods. Depending on τ and when the part was introduced, a part produced in a previous period might be reused multiple times.

We let $v_j(t)$ denote the cumulative production at the start of period j given that the part was introduced in period t. $v_j(t)$ is calculated as follows:

$$v_j(t) = d_{j-1} - q_{j-1}(t) + v_{j-1}(t) \tag{29}$$

where $v_j(t) = 0$ if $t \geq j$. The cumulative production at the start of period j equals the cumulative production at the start of period j-1 plus the amount of new production in period j–1. The amount of production in period j–1 is the difference between the period's demand and the amount of product returned in period j–1.

We assume that the demand in a period is always greater than the amount recycled in the same period. This is a valid assumption if the product is seeing steady annual growth.

11 Algorithm Formulation

As mentioned in Section 10, the formulation for the discrete part selection problem depends on the nature of the performance requirement. This section will treat separately two instances of the performance requirement. In the first case, the performance requirement is deterministic. That is, the performance requirement for each period is known with certainty at the start of period 1. In Section 11.2, this problem is formulated as a shortest path problem. In the second case, each period's performance requirement is an independent random variable. In this case, the distribution for each period's performance requirement is known at the start of period 1, but the period's requirement is not realized until the start of the period. In Section 11.3, this problem is formulated as a backward dynamic program.

Before these three algorithms can be constructed, we first need to describe how the performance requirement and the part performance level interact. This is done in the next section.

11.1 Relating the performance requirement and performance level

Depending on the part being analyzed, the performance requirement can be enforced using a hard constraint or a target constraint. The specifics of the industrial application will dictate which type of constraint will be required. As we will see in later sections, the enforcement technique will significantly affect the problem's structure and the solution procedure. Therefore, we will analyze each of these enforcement techniques separately.

11.1.1 Hard constraint definition

If the enforcement is done using a hard constraint, then the chosen part's performance level must meet or exceed the period's performance requirement. If higher performance values denote superior performance (as in the case of processor speed) then this requires the part's performance level to meet or exceed the performance requirement. If lower levels denote superior performance (as in the case of product weight) then this requires the part's performance level to be no greater than the performance requirement. In either context, there is no penalty for exceeding the period's performance requirement. For example, if a digital camera's CCD must capture 768 bits in the current generation, a CCD capable of 1024 bit resolution is also permissible to use. In this case, the camera's software will ignore the higher resolution and only process images at 768 bits. However, a CCD that can only capture 512 bits is unacceptable because there is no way for this part to capture the required 768 bits.

Thus, the hard constraint acts to further limit the set of candidate parts in period j; only the subset of parts from the set S_j that meet or exceed the performance requirement are candidates when there is a hard constraint.

11.1.2 Target constraint definition

If the enforcement is done using a target constraint, a penalty is imposed based on the deviation of the performance level from the requirement. For example, an example presented later in this chapter utilizes the target constraint when planning the size of a circuit board. The ideal size of the circuit board in the first period is 50 cubic centimeters. Larger and smaller sizes are feasible, but they will require a costly redesign of the product. A target constraint is also applicable when the performance level is an aggregation of several different technical attributes.

Rather than imposing a constraint in the problem formulation, we capture the target constraint by adding a quadratic cost to the objective function. A period-dependent scalar, r_j, is multiplied by the square of the difference between the chosen part's performance level and the realized performance requirement.

11.2 Case 1: Deterministic performance requirement

Let w_j denote the performance requirement in period j. Since the performance requirements are deterministic, their values are known for all $j \in N$ at the start of period 1.

11.2.1 Shortest path formulation

We formulate the shortest path problem on a N+1 node network where the nodes are labeled from 1 to N+1. Each node represents a period in the model, with node N+1 representing the termination of the horizon. An arc from j to k represents selecting a new part in period j and using it through period k–1. In this formulation, a development activity will occur in periods j and k. By construction, there can be no arcs (j,k) such that $j \geq k$.

Let C_{jk} denote the cost on the arc from j to k, namely the cost of choosing the part that is feasible for periods j through k-1 and satisfies the periods' demand requirements at a minimum cost. If there is no part that is feasible for each of the periods j through k-1, then there will be no arc from j to k in the network. C_{jk} is determined below for both the hard constraint and target constraint cases.

11.2.1.1 Hard constraint case[7]

In the hard constraint case, the cost for the arc from j to k equals:

---

[7] For this section we assume that higher performance levels denote superior performance.

$$C_{jk} = \min_{i} \left\{ K + \sum_{\ell=j}^{k-1} \left[\left(1 - e_{i\ell}\left(j, v_{\ell}(j)\right)\right) c_{ij} \left(d_{\ell} - q_{\ell}(j)\right) + y_{i} q_{\ell}(j) \right] \right\} \tag{30}$$

where $1 \le j < k \le N+1$ and i satisfies

$$\left\{ i : i \in \bigcap_{\ell=j}^{k-1} S_{\ell}, p_{\ell}(a_{i}) \ge w_{\ell} \quad \text{for } j \le \ell < k \right\}. \tag{31}$$

Recall that $q_j()$, the number of recycled parts that are available in period j, is given by (28) in section 10.5.2, $v_j()$, the cumulative production in period j, is given by (29) in section 10.5.3, and $e_{ij}()$, the discount rate for part i in period j, is given by (27) in section 10.4.2. Equation (30) represents the minimum cost of choosing a part in period j and using that part through period k-1. There are three components of this cost. First, a development cost of K must be incurred since one development cycle will occur during the interval [j,k–1]. Second, the manufacturing cost for new parts is incurred. The per unit cost in a period is the initial unit manufacturing cost net any discounts accrued since period j. The number of new units manufactured in a period equals the period’s demand net any units that are recycled in the period. Finally, a remanufacturing cost is applied to all of the units that are recycled each period.

The minimization in (30) occurs over the set of parts that meet two conditions: First, they must be available in periods j to k–1. Second, they must meet or exceed each period’s performance requirement. This set is constructed in (31).

11.2.1.2 Target constraint case

As mentioned in Section 11.1.2, the target constraint is captured by adding a quadratic penalty cost to the objective function. The cost of the arc from j to k equals:

$$C_{jk} = \min_{i} \left\{ K + \sum_{\ell=j}^{k-1} \left[\left(1 - e_{i\ell}(j, v_\ell(j))\right)\left(c_{ij}\left(d_\ell - q_\ell(j)\right) + y_i q_\ell(j) + r_\ell\left(p_\ell(a_i) - w_\ell\right)^2\right) \right] \right\} \quad (32)$$

where $1 \leq j < k \leq N+1$ and $i \in \bigcap_{\ell=j}^{k-1} S_\ell$.

The development, manufacturing, and recycling costs in (32) are the same as in (30). The additional term in (32) captures the per period penalty cost incurred when the part's performance level deviates from the period's performance requirement. The only requirement for a part to be considered is that it must be available from period j through period k–1.

11.2.2 Problem complexity

When the shortest path is constructed and solved, the bottleneck operation is the construction of the network itself. Recall that the total number of parts available during the problem's horizon is n. There will be a maximum of N arcs emanating from each node. Therefore, the complexity of the network construction phase is $O(nN^2)$.

11.3 Case 2: independently distributed performance requirements

The performance requirement in period j is a random variable denoted by W_j. At the start of period 1, for all $j \in N$, W_j has a known probability density function $\phi_j(w)$ and distribution function $\Phi_j(w)$. At this point, we assume that the W_j are independent for all $j \in N$. Without loss of generality, we assume that period j's performance requirement, w_j, is realized at the start of period j. At some point before period j's demand occurs, the design team knows the requirement for period j. This model assumes that the requirement becomes known at the start of the period.

11.3.1 Hard constraint case[8]

When the performance requirement is a random variable, the hard constraint case is formulated as a backward dynamic program. There are two state variables: the part used in the previous period and the period in which the part was introduced. We need to keep track of the part's introductory period in order to determine the amount recycled in the current period as well as the current period's discount rate.

Let $g_j(i,t,k)$ denote the cost in period j if the initial state is (i,t) and part k is selected. $g_j(i,t,k)$ is defined below:

$$g_j(i,t,k) = \begin{cases} \left(1 - e_{ij}\left(t, v_j(t)\right)\right) c_{ij} \left(d_j - q_j(t)\right) + y_i q_j(t) & \text{if } k = i \\ K + c_{kj} d_j & \text{o.w.} \end{cases}$$

The value of $g_j(i,t,k)$ depends on whether or not part i is replaced in period j. If part i is not replaced (k equals i) then the part used in period j is the same part that was used in period j-1, and the recycling stream and discounts from previous periods have to be considered. This corresponds to the first expression for $g_j(i,t,k)$. If part i is replaced (k does not equal i), then

---

[8] For this section we assume that higher performance levels denote superior performance.

production in period j is starting from scratch. There are no discounts to apply and no recycled parts available to remanufacture. In this case, the only two costs that are incurred are the development cost and the cost to manufacture the entire period's demand.

We let $f_j(i,t,w)$ represent the minimum cost from periods j through N given that the state at the start of period j is (i,t) and the realized performance requirement in period j is w. $f_j(i,t,w)$ is formulated below::

$$f_j(i,t,w) = \min_k \{g_j(i,t,k) + f_{j+1}(k,j)\}. \tag{33}$$

where k satisfies $k \in S_j$ and $p_j(a_k) \geq w$.

The cost-to-go function, $f_j(i,t)$, represents the minimum cost from periods j through N given that the state at the start of period j is (i,t). The state variable definition allows the cost-to-go function to be separated into the cost in period j plus the optimal cost-to-go for periods j+1 through N. In the hard constraint case, the cost-to-go equation equals:

$$f_j(i,t) = E\{f_j(i,t,w)\} \tag{34}$$

where $1 \leq j \leq N$.

11.3.2 Target constraint case

As with the deterministic performance requirement presented in Section 11.2.1.2, the stochastic performance constraint can be solved as a shortest path problem when the target constraint is employed. Recall that in the target constraint case, a part that belongs to S_j is not excluded in period j if the part's performance level does not exceed the performance requirement. A cost is incurred in the objective function, but it is still possible to use the part.

Therefore, in the target constraint case, each period's set of feasible parts is known at the start of the horizon.

As in the deterministic formulation, we formulate the shortest path problem on a N+1 node network where the nodes are labeled from 1 to N+1. Each node represents a period in the model, with node N+1 representing the termination of the horizon. In the stochastic formulation, an arc from j to k represents the *expected* cost of selecting a new part in period j and using it through period k–1, where the expectation is taken over the performance requirements from period j through k-1.

Let C_{jk} denote the expected cost of choosing the part that is feasible for periods j through k-1 and satisfies the periods' demand requirements at a minimum cost. If there is no part that is feasible for each of the periods j through k-1, then there will be no arc from j to k in the network. C_{jk} is determined below:

$$C_{jk} = \min_{i} \left\{ \underset{W_j,\ldots,W_{k-1}}{E} \left[K + \sum_{\ell=j}^{k-1} \left[\left(1 - e_{i\ell}\left(j, v_\ell(j)\right)\right) c_{ij} \left(d_\ell - q_\ell(j)\right) + y_i q_\ell(j) + r_\ell \left(p_\ell(a_i) - w_\ell\right)^2 \right] \right] \right\} \tag{35}$$

where $1 \le j < k \le N+1$ and i satisfies $i \in \bigcap_{\ell=j}^{k-1} S_\ell$. Recall that $q_j()$, the number of recycled parts that are available in period j, is given by (28) in section 10.5.2, $v_j()$, the cumulative production in period j, is given by (29) in section 10.5.3, and $e_{ij}()$, the discount rate for part i in period j, is given by (27) in section 10.4.2. Equation (35) represents the minimum expected cost of choosing a part in period j and using that part through period k-1. There are four components of this cost. First, a development cost of K must be incurred since one development cycle will occur during the interval [j,k–1]. Second, the manufacturing cost for new parts is incurred. The per unit cost in a period is the base unit manufacturing cost net any discounts accrued since period j. The actual number of new units manufactured in a period equals the period's demand net any units that are recycled in the period. Third, a remanufacturing cost is applied to all of the units that are recycled each period. Finally, a quadratic penalty cost is applied to the deviation of the part's performance level from each of the performance requirements in periods j through k-1.

We will now show that the stochastic formulation can be converted into the deterministic formulation. Certainly equivalence holds when the optimal solution from a stochastic formulation remains the same after the random variables are replaced with their expected values. Their costs will differ by a constant, but the optimal solution is the same.

That certainty equivalence holds in this case can be seen by performing some basic algebraic manipulations of (35). We first separate the quadratic penalty cost as shown below:

$$C_{jk} = \min_{i} \left\{ K + \sum_{\ell=j}^{k-1} \left[\left(1 - e_{i\ell}\left(j, v_{\ell}(j)\right)\right) c_{ij} \left(d_{\ell} - q_{\ell}(j)\right) + y_i q_{\ell}(j) \right] + \underset{W_j,\ldots,W_{k-1}}{E} \left[\sum_{\ell=j}^{k-1} r_{\ell} \left(p_{\ell}(a_i) - w_{\ell}\right)^2 \right] \right\} \quad (36)$$

Since the expectation of the sum equals the sum of the expectations, we can further rewrite (36) as:

$$C_{jk} = \min_{i} \left\{ K + \sum_{\ell=j}^{k-1} \left[\left(1 - e_{i\ell}\left(j, v_{\ell}(j)\right)\right) c_{ij} \left(d_{\ell} - q_{\ell}(j)\right) + y_{i} q_{\ell}(j) \right] + \sum_{\ell=j}^{k-1} \left\{ r_{\ell} \left(p_{\ell}(a_{i})^{2} - 2 p_{\ell}(a_{i}) E\left[w_{\ell}\right] \right) \right\} \right\} + \sum_{\ell=j}^{k-1} \left[r_{\ell} E\left[w_{j}{}^{2} \right] \right] \tag{37}$$

where the last term has been removed from the minimization because it does not depend on i. Finally, if we complete the square, we can rewrite (37) as:

$$\begin{aligned} C_{jk} = \min_{i} \Bigg\{ & K + \sum_{\ell=j}^{k-1} \left[\left(1 - e_{i\ell}\left(j, v_{\ell}(j)\right)\right) c_{ij} \left(d_{\ell} - q_{\ell}(j)\right) + y_{i} q_{\ell}(j) \right] \\ & + \sum_{\ell=j}^{k-1} \left\{ r_{\ell} \left(p_{\ell}(a_{i}) - E\left[w_{\ell}\right] \right)^{2} \right\} \Bigg\} \\ & + \sum_{\ell=j}^{k-1} \left\{ r_{\ell} \left\{ E\left[w_{j}{}^{2} \right] - E\left[w_{j} \right]^{2} \right\} \right\} \end{aligned} \tag{38}$$

The minimization considers four costs: the development cost, manufacturing cost, remanufacturing cost, and quadratic penalty cost. However, the quadratic penalty cost depends only on the expected values of the performance requirements. The final terms (on the third line of (38)) are outside the minimization since they do not depend on i. On the path from j to k, this is the constant by which the deterministic and stochastic formulations differ.

When the deterministic formulation is populated with the means of the periods' performance requirements, the constant by which the stochastic and deterministic formulations will differ equals:

$$\sum_{\ell=1}^{N} \left\{ r_{\ell} \left\{ E\left[w_{j}{}^{2} \right] - E\left[w_{j} \right]^{2} \right\} \right\} \tag{39}$$

The constant by which the two formulations differ equals the summation of each period's penalty cost times the variance of the period's performance requirement. As the variance increases, so too does the constant.

Recasting the stochastic formulation as a deterministic shortest path problem significantly reduces the problem's computational time. Certainty equivalence will always hold when the problem consists of a quadratic penalty function and linear constraints. For a different construction of the certainty equivalence result, the interested reader is referred to Bertsekas (1995).

12 Example

This section presents an exploratory application of the part selection problem. Section 12.1 describes the company's current part selection process. Section 12.2 presents an initial investigation to see if the model is applicable to their problem.

The description of the process in this section comes from a company that is currently investigating the applicability of this solution procedure to their process. Since the company has not yet adopted the procedure, several pieces of the data presented in Section 12.2 are solely estimates on my part. The application is not yet at a point where the required data has been gathered. The data that has been estimated will be noted in the section. Furthermore, to disguise the company's proprietary information, the product in section 12.1 has been heavily disguised. The description of the process is accurate, but the product's context is totally different.

12.1 Current Process

The firm's product competes in a high volume consumer-focused business. For the purposes of this section, we will assume that the firm sells a handheld personal digital assistant (PDA). Size, as measured by the product's volume, is the primary differentiating characteristic in the market. As a general rule, the smaller the product, the more desirable consumers find the product. The product category has existed for several years now and the company has been quite good at estimating the size requirements for future versions of the product.

Figure 12-1 contains a disguised version of the recent history for the product category's volume requirements over time.

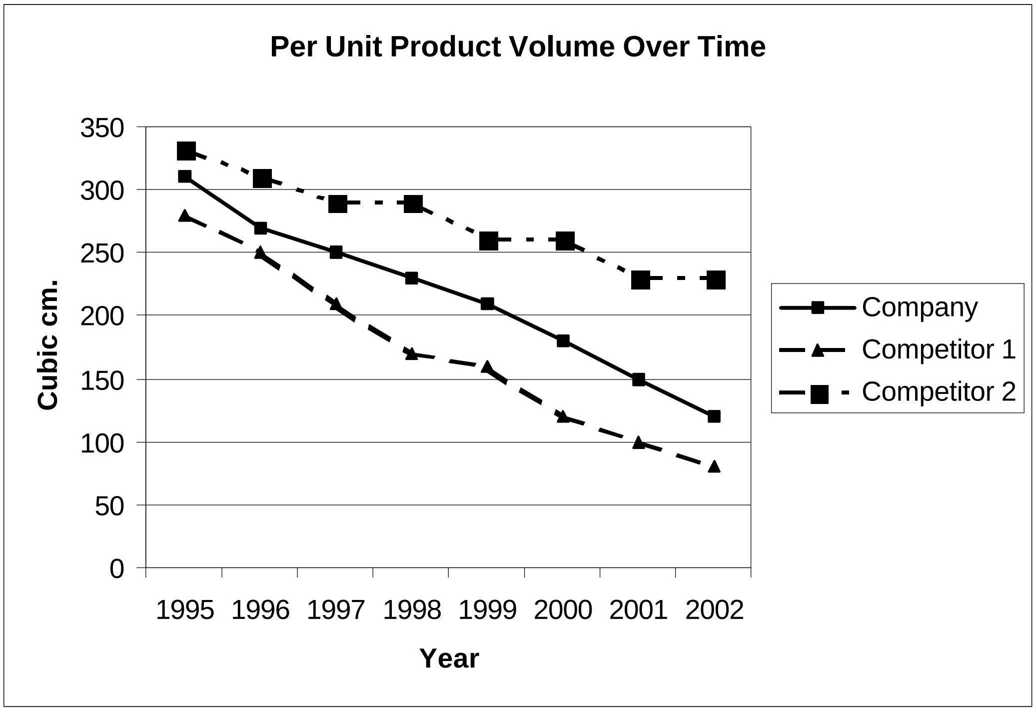

Figure 12-1: Chart showing product volumes over time

The market place is populated with three firms, although it is dominated by two firms: Competitor 1 and the sponsor company. The customer marketplace has been segmented according to size and cost preferences. Competitor 1 has established itself as the leader in PDA miniaturization, allowing it to charge a market premium and to attract lead users who value this feature. The sponsor company has kept pace with Competitor 1 but its product is consistently larger. Its product costs slightly less, thereby targeting a more price sensitive customer segment.

Currently, product introductions occur once a year in the Fall. The product introduction coincides with the industry's large trade show.

A PDA is composed of several standard off-the-shelf components like the LCD display and the serial connector. The company's operating procedure has been to assume that each component takes up a certain percentage of the product's total volume. Smaller and larger volume options are available, but the smaller volume option will significantly increase the per unit cost of the component. The bulk of the cost for the product is the circuit board. Although this part's functionality does not change significantly from year to year, its size does.

As a general rule, the circuit board consumes twenty five percent of the product's volume. If a higher volume circuit board is chosen, then less space is available for other parts, so more expensive components will have to be used for the other parts. If a lower volume circuit board is chosen, then the plastic housing can be designed to fill the space thatis not consumed by the circuit board. However, a smaller circuit board requires smaller parts which can significantly increase the board's price.

Each generation, the cost of a new circuit board with a volume of 25% of the product's volume is typically right around fifty cents. Two factors contribute to this general rule. First, the entire industry has a good sense of what the product size requirements will look like over time. Second, the industries that provide components are also miniaturizing their offerings at a well-defined rate. The combination of these two factors means that the price for a standard set of components is nearly constant from generation to generation, although the size of the components has decreased each generation.

If a firm wanted to use a smaller circuit board, that would be possible but the per unit cost can run upwards of one dollar and fifty cents. In this firm, one often hears the comment "pennies matter." And this advice is quite appropriate. Since product volumes are 400,000 units a year and growing at 15% a year, the cost of choosing a circuit board that is smaller than necessary can easily exceed one hundred thousand dollars in the first year.

An important part of the company's strategic vision is the ability to let customers return a previous generation's product in exchange for a discount towards the purchase price of a new product. This corporate strategy was put in place in order to decrease the likelihood that an existing customer will switch to a rival's product. If the returned product's circuit board is the same board that is being used, then it can be put into the current generation's PDA. If the circuit board is a different board, then the entire product is scrapped. The circuit board is the only part worth recycling. All of the other components have either been scratched or are incompatible with the current product's design. While it is true that the plastic parts from each returned product are pulverized into plastic pellets, this cost is such a small part of the product's cost that it can safely be ignored.

The company's design strategy has been to redesign the product from scratch each generation (each year). This means that all components, including the circuit board, are redesigned each year. There are several reasons the company has done this. First, the company is struggling to revise the product each generation. There are few resource available to devote to solving problems that do not concern the next product launch. Second, the entire customer segment is growing rapidly. Therefore, the company's emphasis has been on delivering products that the customer wants to purchase in the current year. This product has not yet seen the kind of cost consciousness that is more prevalent in mature market segments. Third, up to this point, the company has had no way to characterize the benefits of doing anything other than myopically optimizing each generation's design.

To summarize, the company's current circuit board selection process consists of choosing the part that minimizes the circuit board cost in the current generation. This involves minimizing the per unit cost plus any cost that must be incurred based on the board's deviation from the ”ideal” size that period. Whether a smaller or larger part is chosen will depend on the relative importance of these two costs.

12.2 Handheld Product Example

To explore the utility of this approach, this section will examine the part selection process for the years 1999 to 2002. The circuit board's yearly performance requirement is displayed below in Table 12-1:

Year	Performance Requirement (cm^3)
1999	52.5
2000	45.0
2001	37.5
2002	30.0

Table 12-1: Circuit Board Performance Requirement

The performance requirement is estimated to equal twenty-five percent of the product's performance requirement; recall that the company's estimate of the performance requirements were displayed in Figure 12-1. We assume that the performance requirements are deterministic.

We assume that there are 8 different circuit boards available. For simplicity, all eight boards are available in every period. The boards have performance levels ranging from 54 to 26 cubic centimeters. Although the performance level of each board does not vary by period, its initial part cost does vary. We assume that the board's per unit cost in a period is derived from the equation:

$$\min\{ \$.50 + (w - p) \times \$.03 , \$.33 \}$$

where w is the period's performance requirement and p is the part's performance level.

Although these numbers are entirely fictitious, the formula captures the proper behavior of a part's unit cost. If the part performance level equals the requirement, then the part costs fifty cents. If the part's performance level is larger than the requirement, then the part costs less than fifty cents. This corresponds to the case where the part is an older part. However, no part can cost less than 33 cents. Finally, if the part's performance level exceeds (is less than) the performance requirement, then the part is more expensive. The higher expense could either reflect the scarcity of the part or the increased cost due to the smaller components used. The performance level and initial part cost per period for each part are shown in Table 12-2:

		Initial Cost by Year			
Part	Performance Level	1999	2000	2001	2002
1	54	0.46	0.23	0.33	0.33
2	50	0.58	0.35	0.33	0.33
3	46	0.70	0.47	0.33	0.33
4	42	0.82	0.59	0.37	0.33
5	38	0.94	0.71	0.49	0.33
6	34	1.06	0.83	0.61	0.38
7	30	1.18	0.95	0.73	0.50
8	26	1.30	1.07	0.85	0.62

Table 12-2: Part Performance Levels and Initial Costs

The development cost is initially assumed to equal $250,000 whenever the circuit board's design is modified. The entire product will be revised each period but the circuit board will only be revised when necessary. For this reason, the rest of the product's development costs are viewed as fixed and the analysis in this section focuses on the development of the circuit board.

In order to capture the effect of the recycling process, each year is separated into two six-month periods. Therefore, there are eight periods in the model. The recycling rate in the period after the part's introduction is assumed to be 50%. In all subsequent periods, the recycling rate is zero; using the notation in Section 10.5.2, $\tau = 1$.

To achieve the stated annual growth rate of 15%, the semiannual growth rate is set at 7.24%. As a starting point, we also assume that the demand in the first half of 1999 equals 200,000 units.

There is no time discount but there is a 1% quantity discount for every 100,000 circuit boards that are produced.

The scalar associated with the quadratic penalty cost was set to equal 1000. Recall that this scalar will be multiplied by the squared difference between the part's performance level and the period's performance requirement. This number is entirely an estimate on my part. The number was determined by recognizing that if the part and the performance requirement differ by more than 10 cubic centimeters, then there will be a significant redesign of the PDA's other components. Another way to view this scalar is that, each period, it acts to limit the set of parts to some set of parts centered at the ideal part performance level. Setting this scalar higher will act to reduce the set of parts that are realistic to consider each period.

The results in the table below summarize the optimal solution and the company's heuristic of redesigning the circuit board each period:

Company Heuristic

Part Used	Period Introduced	Last Period Used	Total Cost
1	1999	1999	416,545
3	2000	2000	443,737
4	2001	2001	467,646
6	2002	2002	492,179
		Total ($k)	1,820

Optimal Upgrade Strategy

Part Used	Period Introduced	Last Period Used	Total Cost
2	1999	2000	697,595
5	2001	2002	819,782
		Total ($k)	1,517

Table 12-3: Cost Summary

The company's heuristic is approximately $303,000, or 20%, more expensive than the optimal upgrade strategy. Recall that the arc from i to j denotes a part revision in both years i and j. We define a part selection strategy as a sequence of parts that cover every period in the model. That is, each period is served by exactly one part. To better understand the results in Table 12-3, we can separate the total cost for a part selection strategy into three costs: development cost, production cost and conformance cost. Development cost is the sum of the development costs incurred over the problem's horizon. The production cost is the total manufacturing and remanufacturing costs over the production horizon. The conformance cost is the total penalty costs incurred over the horizon. A summary of these costs is displayed in Table 12-4:

Company Heuristic

Part Used	Period Introduced	Last Period Used	Development	Production	Conformance
1	1999	1999	250000	162045	4500
3	2000	2000	250000	191737	2000
4	2001	2001	250000	177146	40500
6	2002	2002	250000	210179	32000
		Totals ($k)	1000	741	79

Optimal Upgrade Strategy

Part Used	Period Introduced	Last Period Used	Development	Production	Conformance
2	1999	2000	250000	385095	62500
5	2001	2002	250000	441282	128500
		Totals ($k)	500	826	191

Table 12-4: Detailed Cost Analysis

The company's heuristic does a good job of minimizing production and conformance costs, but it must incur a large development cost to achieve this. The optimal upgrade strategy incurs higher production and conformance costs but offsets these costs by skipping two product development cycles. The optimal strategy uses each circuit board for two product generations. To accomplish this, the optimal strategy overdesigns the product in the first year, incurring extra manufacturing cost. In contrast, the company's heuristic "underdesigns" the part each generation. Since the company is redesigning the part each generation, it is clear that they will either choose between the two parts that are closest to the performance requirement. Given the parameters presented in this example, the optimal decision is to choose the larger volume part. This is due to the decreased manufacturing costs associated with older parts.

At this point, it is difficult to come up with a mathematical condition for the optimal number of revisions over the product horizon. However, several general insights can be drawn from an examination of the problem's structure. These results are summarized in Table 12-5 and described below in more detail.

Variable	Action	Effect on Number of Revisions
Development Cost	Increase	Decrease
Demand	Increase	Increase
Penalty Cost	Increase	Increase
Recycling Rate	Increase	Decrease

Table 12-5: General Behavior of Optimal Solution

As the development cost increases, the optimal number of revisions decreases. This is due to the fact that if the development cost increases, then it is more attractive to find paths that increases the penalty and conformance cost while lowering the development cost. As demand increases, the number of revisions also increases. The primary cost involved here is the production cost. As demand increases, the production cost increases. An increase in the penalty cost increases the number of part revisions. Increasing the penalty cost increases the total penalty cost, making it more optimal to find parts that might require a higher manufacturing cost but are more closely aligned with each period's requirement. Finally, increasing the recycling rate decreases the number of revisions. Increasing the recycling rate decreases the production cost when parts are used longer.

An informal way to look at these results is to note that different variables affect the self-sufficiency of each generation. Conditions that promote self-sufficiency in each generation make it cost effective to redesign the circuit board more often.

13 Conclusion and Next Steps

This dissertation has presented two models in supply chain design. The common theme of both of these projects is that they develop a quantitative model that moves beyond solely considering UMC when sourcing a supply chain.

For the supply chain configuration problem, there are several relevant extensions to consider. First, time-to-market costs should be incorporated. This can be accomplished by augmenting the two-state formulation with a third state variable. The additional state variable is the maximum replenishment time. Once the state variable has been added, the time to market cost can then be applied as a function of the stage's maximum replenishment time. Second, the current algorithm's implementation needs to be refined. Implementing an efficient data structure should significantly improve the algorithm's performance. Third, one-state approximations need to be developed. Since the two-state variable approach is significantly more computationally burdensome than the one state-variable problem, this will be an important area to study in the near future. Finally, it would be worthwhile to consider incorporating side constraints that will make the model more applicable in practice. The constraint that immediately comes to mind is a limit on the number of different vendors that the model can choose. Although I have not yet developed a solution technique to address these kinds of constraints, I suspect that the most fruitful approach will be to preprocess the available options at a stage, pruning some set of options that would violate the side constraint. If this is done as a preprocessing step, then the original solution procedure will still be valid.

There are also several interesting extensions to the part selection problem. First, it will be very useful to extend the model to incorporate an entire product family. The single product assumption is very limiting because it does not capture the role of downgrading in the product development process. When downgrading is present, all but possibly the lowest-ranked part will still be available in the next period. This model would address how many different parts to offer and the schedule for when new parts should be introduced. The multiproduct model would also be useful in determining the level of modularity that a product should possess. If there are multiple products in the product family, then a modular part could be used in several of these products. However, if an integral design is chosen, then each part will likely have to use its own unique part. Second, it would be useful to introduce more general penalty cost functions. A more applicable penalty cost function would impose no penalty cost if the requirement is exceeded but a quadratic (or more general) cost would be to parts that do not meet the performance requirement. For these more general cost functions, certainty equivalence would not hold. Finally, the performance requirement should capture correlation over time. That is, the performance requirement in the current period should be no less than the performance requirement in the previous period.

Finally, as a longer-term goal, it would be very useful to integrate these two models. In fact, when I first began this research I viewed these two models as one unified model. However, after some initial discussion with industry it became clear that these two models are currently targeted at two different constituencies within the product development process. The supply chain configuration work is directly applicable to the materials manager that is charged with sourcing a new supply chain. The part selection problem is relevant to the designer that is choosing what options to include in the product. In most companies, these two groups do not communicate. Therefore, the challenge in integrating these two models will have as much to do with getting an organization to change it product development process as it will have to do with getting the mathematics correct.

14 References

Bertsekas, D. P.., Dynamic Programming and Optimal Control, Belmont, Athena Scientific, 1995.

Chand, S., and S. Sethi, "Planning Horizon Procedures for Machine Replacement Models with Several Possible Replacement Alternatives," *Naval Research Logistics Quarterly*, 29 (1982), 483-493.

Clausing, D., Total Quality Development : a Step-by-Step Guide to World Class Concurrent Engineering, New York, ASME Press, 1993.

Cohen, M. A. and R. H. Halperin, "Optimal Technology Choice in a Dynamic-Stochastic Environment," *Journal of Operations Management*, 6 (1986), 317-331.

Cohen, M. A., J. Eliashberg, and T.-H. Ho, "New Product Development: The Performance and Time-To-Market Tradeoff," *Management Science*, 42 (1996), 173-186.

Datar, S., C. C. Jordan, S. Kekre, S. Rajiv, and K. Srinivasan, "Advantages of Time-Based New Product Development in a Fast-Cycle Industry," *Journal of Marketing Research*, 34 (1997), 36-49.

Ettl, M., G. E. Feigin, G. Y. Lin and D. D. Yao, "A Supply Network Model with Base-Stock Control and Service Requirements," IBM Technical Report (RC 20473), 43 pages, 1996. Submitted to *Operations Research*.

Fine, C. H. and R. M. Freund, "Optimal Investment in Product-Flexible Manufacturing Capacity," *Management Science*, 36 (1990), 449-466.

Graves, S. C., "Safety Stocks in Manufacturing Systems," *Journal of Manufacturing and Operations Management*, 1 (1988), 67-101.

Graves, S. C., and S. P. Willems, "Optimizing Strategic Safety Stock Placement in Supply Chains," Working Paper, 49 pages, January 1998.

Grieco, P. L., Jr., and M. W. Gozzo., Supplier certification II : handbook for achieving excellence through continuous improvement, Palm Beach Gardens, PT Publications, 1992.

Hendricks, K. B., and V. R. Singhal, "Delays in New Product Introductions and the Market Value of the Firm: The Consequences of Being Late to the Market," *Management Science*, 43 (1997), 422-436.

Inderfurth, K., “Valuation of Leadtime Reduction in Multi-Stage Production Systems,” in: G. Fandel, T. Gulledge and A. Jones (eds.), Operations Research in Production Planning and Inventory Control, Springer, Berlin, 1993, 413-427.

Kimball, G. E., “General Principles of Inventory Control,” *Journal of Manufacturing and Operations Management*, 1 (1988), 119-130.

Krishnan, V., R. Singh, and D. Tirupati, “A Model-Based Approach for Planning and Developing a Family of Technology-Based Products,” to appear *in Manufacturing & Service Operations Management*, 37 pages, 1998.

Laseter, T. M., Balanced Sourcing : Cooperation and Competition in Supplier Relationships, San Francisco, Jossey-Bass Publishers, 1998.

Lee, H. L., and C. Billington, “Material Management in Decentralized Supply Chains,” *Operations Research*, 41 (1993), 835-847.

Nurani, R. K., S. Seshadri, and J. G. Shantikumar, “Optimal Control of a Single Stage Production System Subject to Random Process Shifts,” University of California at Berkeley Working Paper, February 1994, revised February 1995.

Rajagopalan S., Singh M. R., and Morton T. E., "Capacity Expansion and Replacement in Growing Markets with Uncertain Technological Breakthroughs," *Management Science*, 44 (1998), 12-30.

Sanderson, S. W. “Cost Models for Evaluating Virtual Design Strategies for Multicycle Product Families,” *Journal of Engineering and Technology Management*, 8 (1991), 339-358.

Simpson, K. F., “In-process Inventories,” *Operations Research*, 6 (1958), 863-873.

Terborgh, G., Dynamic Equipment Policy, New York, McGraw-Hill, 1949.

Ulrich, K., D. Sartorius, S. Pearson and M. Jakiela, “Including the Value of Time in Design for Manufacturing Decision Making,” *Management Science*, 39 (1993), 429-447.