Abstract

BRYKSINA, ELENA ALEXANDROVNA. Assessing the Impact of Strategic Safety Stock Placement in a Multi-Echelon Supply Chain. (Under direction of Dr. R. Handfield and Dr. D. Warsing)

The objective of this study is to develop prescriptions for strategically placing safety stocks in an arborescent supply chain in which there are moderate to severe risks of disruptions in supply. Our work builds off of recently published work by Graves and Willems (2003) that demonstrates that a simple-to-compute, congestion-based adjustment to supply lead times, first developed by Ettl et al. (2000), can be embedded in a non-linear optimization problem to minimize total investment in safety stock across the entire supply chain. We are interested in investigating how the Graves and Willems (GW) model performs under uncertainty in supply. We first propose an adjustment to the model (Mod-GW) by considering two types of fulfillment times, a normal fulfillment time L_j and a worst possible fulfillment time K_j, which allows us to account for supply uncertainty, or disruptions in supply. We evaluate the performance of GW and Mod-GW using Monte Carlo simulation and, using motivation from Timed-Petri Net analysis, develop an Informed Safety Stock Adjustment (ISSA) algorithm to compute the additional buffer stock levels necessary to improve downstream service performance to the target level. We find that the service performance of the Mod-GW solution is most sensitive to the probability of disruption at any node in the supply chain, requiring higher safety stock adjustments through ISSA as this probability increases. In particular, the relative value of the holding costs for components and finished goods—and the resulting impact on where safety stock is held in the network—is an important moderating factor in determining the level of service performance degradation of the Mod-GW solution as either p_j, the probability of

disruption at node *j*, or K_j / L_j, the ratio of the disrupted and normal lead times, increases (i.e., as disruptions exert more impact on the network). The Informed Safety Stock Adjustment algorithm generally suggests a sufficient complementary amount to the safety stock.

ASSESSING THE IMPACT OF STRATEGIC SAFETY STOCK PLACEMENT IN A MULTI-ECHELON SUPPLY CHAIN

By

ELENA ALEXANDROVNA BRYKSINA

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

OPERATIONS RESEARCH

Raleigh

2005

APPROVED BY:

____
Chair of Advisory Committee

____
Co-Chair of Advisory Committee

____
Committee Member

____
Committee Member

Biography

Elena A. Bryksina was born in May 1980 in Russia. She attended Methodist College in Fayetteville, NC, from August 1999 to December 2002 and graduated with a Bachelor of Science degree in Business Administration, Marketing, and Mathematics as an honor student.

In August 2003, she entered North Carolina State University in Raleigh, NC, to pursue a Master of Science degree; majoring in Operations Research with a minor in Business. She is currently working as a Forecast Analyst at Lowe's Home Improvement, Inc.

Acknowledgement

I wish to express gratitude to those who have supported and assisted me in writing my MS thesis. Very special thanks and appreciation are due to Dr. Donald Warsing, for his guidance, encouragement and mentoring. I feel privileged to have had him for an advisor. To Jennifer Blackhurst for her motivation and enthusiasm in the research. To Robert Handfield, and Michael Kay, I extend my appreciation for their valuable input into my research.

Table of Contents

List of Tables

List of Figures

1. Introduction

The subject of managing risks in supply chains is gaining a significant amount of notoriety in Operations Management, Management Science and Operations Research. A recent issue of *Production and Operations Management* journal (2005, v 14, n 1) was fully devoted to the subject. It is not clear, however, that practicing managers have a sufficiently well-developed sense of how to use safety stocks to mitigate risks—stemming both from uncertainty in demand and uncertainty in supply—in a manner that is both effective in accomplishing the risk mitigation *and* efficient in not driving down profits. A College of Management professor at NCSU frequently uses the example of a major U.S. auto manufacturer whose risk mitigation strategy consisted of mandating that its plants, it suppliers, and its suppliers' suppliers all carry six weeks' worth of inventory across all items (Blackhurst, 2005). Clearly, such a "one-size-fits-all" approach to risk mitigation is unlikely to be an efficient approach. On the other extreme there are examples where not enough inventory was left in the system to reduce customer response times, and increase agility (Parker 2000). Not understanding the use of inventory in buffering against uncertainty can leave companies handicapped in the event of a disruption in supply or production. Karpinski (2001) reports that prior to the September 11, 2001, terrorist attacks in the U.S., companies did not typically account for the possibility of extensive and/or long disruptions in their inventory planning. Thus, there is a definite need among companies to know how to plan for disruptions.

The author's experience in talking to practicing managers about where to place safety stock and at what levels lends support to the notion that the anecdotal examples above are not unique and that managers generally are searching for reasonably straightforward, but still effective and efficient, approaches to mitigating risks. The existing business literature suggests some approaches for risk mitigation that are more general but not mathematically rigorous (Chopra, 2004; Anonymous, 2003). On the other extreme, there are many complex mathematical models that follow very strict assumptions, limiting their applicability in practice (Wee, 2004). Hence, there is a need for a methodology that will balance these two extremes of generality and specificity. The desired methodology would, on the one hand, provide a company with some guiding principles, but on the other hand, have some analytical rigor supporting those guidelines.

The rest of the paper is organized as follows. Section 2 outlines the objectives of the study, and Section 3 provides a review of the literature that is relevant to this research. Section 4 states the proposed model and methodology, and Section 5 presents our computational experiments, analysis, and discussion of the results. Finally, Sections 6 and 7 provide conclusions and recommendations for future research.

2. Objectives

The objective of this study is to develop prescriptions for strategically placing safety stocks in an arborescent supply chain in which there are moderate to severe risks of disruptions in supply. Our approach is to take an existing model for safety stock placement in a supply chain network (Graves and Willems, 2003), adjust this model to account for disruptions in supply and/or production, and develop a simulation-based, algorithmic approach to modifying it further where necessary. The "where necessary" portion of the previous sentence leads us to generate the strategic prescriptions for safety stock placement.

3. Literature Review

In this section we would like to present different supply chain modeling methods that are available today. Since our model, presented later in the paper, is a combination of two distinct modeling techniques we feel that it is beneficial for the reader to have an overview of the work done in the past in the area.

3.1 Comprehensive Supply Chain Models

Quantitative models for supply chain management can be grouped into three broad classes: analytical performance models, simulation models, and optimization models (Tayur, 1999). A comprehensive overview of existing tools and techniques used to solve supply chain problems is presented by Hicks (1997).

3.1.1 Analytical performance models

Models of supply chains in stochastic environments typically consider the network as a discrete-event, dynamic system. Such systems can be studied as Markov chains, stochastic Petri nets, or queuing network models (Raghavan, 1998; Viswanadham, 1992). Malone and Smith (1988) have looked at organizational and coordination structures, which constitute a key element of any business process. Raghavan and Viswanadham (1998) discuss performance modeling and dynamic scheduling of make-to-order supply chains using fork-join queuing networks. Viswanadham and Raghavan (1998) compare make-to-stock and assemble-to-order systems using generalized stochastic Petri net models. They also use integrated queuing and Petri net models for solving the decoupling point location problem - i.e., the point (facility) in the supply chain at which all finished goods are assembled to confirmed customer orders.

3.1.2 Simulation models

The models discussed above are highly abstracted models of business processes that require significant simplifying assumptions to allow their formulation and solution. To obtain a very accurate and detailed model, one has to represent many realistic features, which may only be possible through the use of a simulation model. Simulation models for supply chain decision making have become prevalent in the literature in recent years. Examples of such studies can be found in Malone (1987), Connors et al. (1996), and Feigin et al. (1996). A “combination study,” of sorts, is presented by Bhaskaran and Leung (1997), who describe re-engineering of supply chains using queuing network models and simulation. Feigin et al. (1996) have looked into enterprise modeling and simulation in an object oriented environment. Similar work has been done by Mujtaba et al. (1994) and Chu (1997). Swaminathan et al. (1997) built-on the work of Mujtaba et al. (1994) and Chu (1997) by using a set of generic objects representing various supply chain entities. Using a generic object-based agent framework, they demonstrate how software objects can be used to build simulation models for a variety of supply chain networks.

3.1.3 Optimization models

One major focus area of supply chain optimization models is to determine the location of production, warehousing, and sourcing facilities, and the paths the products take through them. These methods provide models mostly for strategic and strategic/tactical levels. One of the earliest and most often cited works in this area is that of Geoffrion and Graves (1984). They describe a mixed integer programming model for determining the location of distribution facilities. Along similar lines, Cohen and Lee (1988, 1989) consider global manufacturing and distribution networks and formulate mixed integer optimization programs.

Another significant portion of the supply chain literature consists of multi-echelon inventory control models. A comprehensive review of these models can be found in Vollman et al. (1997). These methods generally deal with operational or tactical/operational levels. Such multi-echelon inventory models have been successfully implemented in industry. For instance, Billington and Lee (1993) develop a multi-echelon inventory model to reflect the decentralized supply chain witnessed in Hewlett-Packard's DeskJet printer supply chain. They develop a single-stage base-stock model that uses target service levels as its inputs. They develop a single-stage, base-stock model that uses target service levels as its inputs. Their model is a tractable approximation that can be expanded into multiple stages. Ettl et al. (2000) take the work of Billington and Lee and put it in an optimization context where the goal is to minimize the overall inventory capital and guarantee the customer service requirements.

3.2 Models for Safety Stock Placement

Our work builds most directly off of recently published work on developing effective heuristically-based computational rules for placing safety stocks in arborescent supply chains subject to uncertainty in demand and, in essence, congestion in supply. Most recently, Graves and Willems (2003) demonstrate that a simple-to-compute, congestion-based adjustment to supply lead times, first developed by Ettl et al. (2000), can be embedded in a non-linear optimization problem to minimize the total investment in safety stock across the entire supply chain. The motivating work behind these approaches was done by Lee and Billington (1993).

While finding the inventory-minimizing safety stock levels based on the lead-time-inflation heuristic of Ettl et al. (2000) can provide some insights into safety stock placement, it makes two assumptions that we consider to be restrictive and not sufficiently reflective of the real value of risk mitigation as viewed by many practitioners—especially in the wake of the 9/11 terrorist attacks, which not only crippled supply chains for a brief period of time in 2001, but more importantly, caused managers and supply chain academics alike to consider the importance of the "resiliency" of a supply chain to minor, moderate, or major disruptions in supply. The two restrictive assumptions are (1) that "nominal" supply lead time at each node is a constant (although congestion-based lead time *is* a random variable) and (2) that unmet demand in a given period is either fully backlogged or fully lost. In this thesis, we focus only on relaxing just first restrictive assumption. Relaxing the second assumption is left for future research.

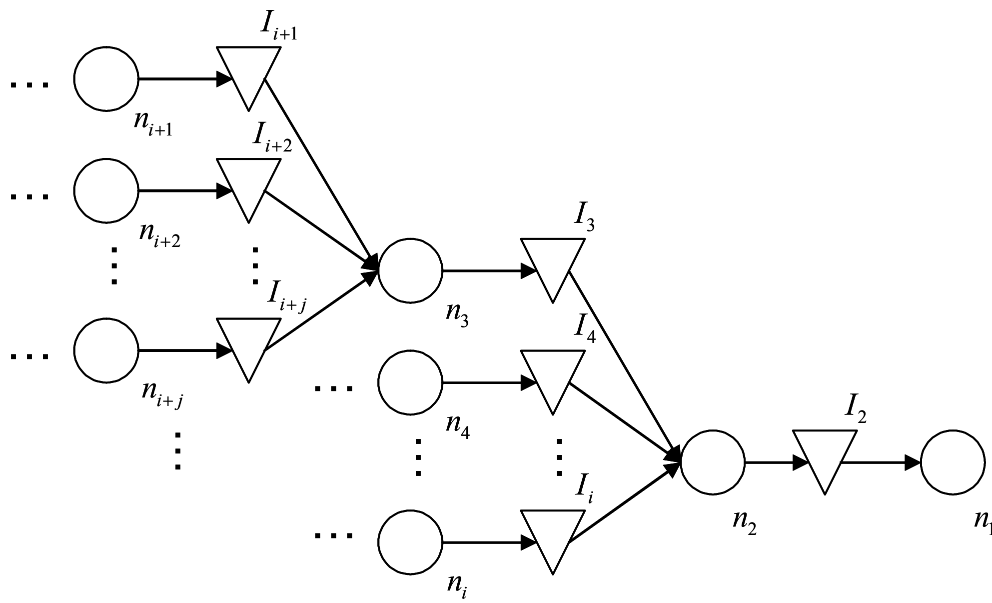
4. Model and Methodology

We use the Graves and Willems (2003) model as our base model and modify it to better fit our research. The Graves and Willems model follows a periodic review policy. In our implementation, inventory is reviewed every period. Each stage of the supply chain operates according to a base-stock policy, placing a replenishment order on its suppliers equal to the observed demand each period. The suppliers have no capacity constraints. The model assumes that per-period demand has mean μ, and standard deviation σ. The demand is stationary and independent for non-overlapping periods. The goal of the model is to determine the inventory level at each node of the supply chain to meet a target service level—i.e., a target probability of not running out of stock before replenishment arrives. The target service level for the external customer is an exogenous variable dictated by the market. The service targets for the internal customers are decision variables in our optimization context.

4.1 Base Model (Graves and Willems Model)

Our supply chain network is defined as a graph $G = (N, E)$, comprised of a set of nodes $N = \{1, \ldots, n\}$ and edges (arcs) $E = \{(i, j) : i, j \in N, i \neq j\}$. Figure 1 shows the generic, n – node supply chain, where node 1 (n_1) is the final customer, and inventory I_i is controlled by node n_i.

Figure 1 – Generic n-node Supply Chain



We define the nominal amount of time required by any node *j* to fulfill an order as the *fulfillment time*, which we denote by L_j, and we denote the *replenishment time* of node *j* by τ_j. Thus, as indicated by Graves and Willems (following the model of Ettl et al. 2000), the worst-case replenishment time of node *j* is

$$\tau_j^{\max} = L_j + \max_{i:(i,j)\in E} \{\tau_i\}, \tag{1}$$

but the expected replenishment time can be expressed as

$$E\left[\tau_j\right] = L_j + \sum_{i:(i,j)\in E} \pi_{ij} L_i, \tag{2}$$

where π_{ij} is the probability that supply node *i* causes a stock-out at *j* and can be estimated using expression (3.4) in Graves and Willems (2003), which comes originally from Ettl et al. (2000). Using these expressions—and assuming that demand over the replenishment lead time at each node *j* is normally distributed with mean $\mu_j E\left[\tau_j\right]$ and standard deviation $\sigma_j \sqrt{E\left[\tau_j\right]}$ — leads to a relatively straightforward non-linear optimization problem to minimize the total investment in safety stock across the network. Specifically, that optimization problem is as follows:

$$\min\ C = \sum_{j=1}^{N} h_j \sigma_j \sqrt{E\left[\tau_j\right]} \left(k_j + \int_{k_j}^{\infty} \left(z - k_j\right) \phi(z)\, dz \right), \tag{3}$$

subject to

$$k_1 = \Phi^{-1}(\alpha),$$

where α is the target service level at the final customer, Φ is the standard normal cdf, ϕ is the standard normal pdf, h_i is the annual inventory holding cost at node *i*, and the other parameters are as defined above. The safety stock factors k_i ($i = 2, \ldots, n$) are the decision variables.

4.2 Methodology

We are interested in investigating how the Graves and Willems (GW) model performs under uncertainty. The model presented above addresses only one aspect of variability, demand uncertainty. We are curious to know how the GW model will perform once supply variability is added to the environment. We anticipate that the above model will not yield the desired results when both types of uncertainty are present. We say the model performs well if the target service level is reached at the final node. By considering two possible fulfillment times, a normal fulfillment time L_j and worst possible fulfillment time K_j, we can account for supply uncertainty, or disruptions in supply. Our supply disruption model is fully discussed in Section 4.3. Finally, if accounting for disruptions by modifying fulfillment times is not sufficient, we propose an Informed Safety Stock Adjustment (ISSA) algorithm, addressed in Section 4.4, to bridge the gap between the actual and the target service levels.

To accomplish our research goal we propose to use Monte-Carlo simulation. Simulation can assist us in creating a model of the operation of the supply chain and allow us to conduct a series of numerical experiments to gain better understanding of the behavior of the chain under supply uncertainty, and more importantly, a better understanding of where additional need of safety stock is needed. We compute modified base-stock levels by altering the GW model to account for supply uncertainty—resulting in what we call the Mod-GW model—and finally, we apply our ISSA algorithm.

We designed our simulation model in Visual Basic. The model simulates 1000 periods of a given chain operations and uses 10 runs to obtain average results of the key performance measurements. The inputs for the model are the fulfillment times, annual holding costs, probability of a disruption at each node, and the solutions for the base stock levels from the Graves and Willems model. The outputs from the simulation are average service level for each node, average ending inventory at each node, average fill rate for each node, average disruption count for each node, average number of times a downstream node was impacted by a disruption at an upstream node, average disruption length at each node, and average disruption length experienced by a downstream node as a result of a disruption at an upstream node.

4.3 Modeling Supply Chain Disruptions (Modified Graves and Willems Model)

Recall that $K_j > L_j$ is the "worst-case" fulfillment lead time (i.e., the "disrupted" lead time) for node *j*, and let p_j be the probability that node *j* is disrupted in any period *t*. Since expression (2) above is itself an estimate, we merely extend the estimate further by using $\left(1-p_j\right)L_j + p_j K_j$ in place of L_j; solving objective function (3) with this update yields the Mod-GW solution.

For the purposes of our Monte Carlo simulation, we define the *disruption state* in period *t* of node *i* to supply node *j* as

$$DS_{i,j,t} = \begin{cases} 1 \text{ if } (i,j) \text{ supply is disrupted in period } t \\ 0 \text{ if } (i,j) \text{ supply is not disrupted period } t. \end{cases} \quad (4)$$

Let $r_{j,t}$ be the "remaining disruption time" in period *t* of node *j*. When a disruption occurs, $r_{j,t}$ will initially be assigned the sampled value of a random variable describing the disruption time, denoted by $DT_{i,i}$, which follows a probability distribution defined on the interval $\left[1, K_j - L_j\right]$. For each succeeding period, $r_{j,t}$ will decrease by one unit, until it reaches zero. In addition, an order placed on node *j* by node *i* at *any* period *t* is fulfilled with lead time $L_j + r_{j,t}$.

Thus, the sampling process to determine if a node *i* is disrupted from supplying node *j* in a given period *t* is as follows:

for $i = 1$ to n

$DS_{i,j,t} = 0$

% start by assuming node i will be "up" this period

for all *j* such that $(i,j) \in E$

$r_{i,t} = \max\{0, r_{i,t} - 1\}$

% remaining disruption time decreases by one, or disruption is over

if $DS_{i,j,t-1} = 0$ then

% if node i was "up" last period, then draw to determine if node i is to be

% disrupted this period

Sample p = RAND()

if $p \le p_i$ then

$DS_{i,j,t} = 1$

Sample $r_{i,t}$ from the distribution of $DT_{i,j}$

% node i will be "down" for the succeeding $r_{j,t}$ periods

end if

else

if $r_{i,t} > 0$ then $DS_{i,j,t} = 1$

% if node i was "down" last period, then it stays down if $r_{i,t} > 0$

end if

next j

next i

4.4 Computing Disruption Impact on Supply

Developed by Carl Petri in the 1960s (Petri 1962), the Petri net is a mathematical modeling tool used to graphically represent and analyze complex systems. We can model our supply chain as a Petri Net. The particular aspect of Petri net modeling that we are interested in modifying is *reachability analysis*. As ground work for our development, we use the type of reachability analysis described by Blackhurst et al. (2004), which is used for analyzing the impacts of supply uncertainty on the ability of a supply chain to deliver product after an unexpected event occurs. Though we use the reachability analysis of Blackhurst et al. (2004) as motivation, we introduce time elements and measure the impact of each disruption on the downstream nodes in terms of time units. We describe this process as *timed reachability analysis* (TRA).

Let us define $P_{i,m}^{j} = \{i, k, \ldots, l, m \in N : (i,k), \ldots, (l,m) \in E\}$ as the j^{th} path from node i to m, where a path is a sub-set of E that represents a sequence of arcs leading from node i to m, such that the ending node in the preceding edge is the first node in the consequent arc. Further, we define $Pcount_{i,m}$ to be the number of paths leading from node i to node m. Let

$$ST = \begin{bmatrix} ST_{11} & . & . & ST_{1n} \\ . & & & . \\ . & & ST_{ij} & . \\ ST_{n1} & . & . & ST_{nn} \end{bmatrix},$$

where $n = |N|$, be a matrix whose elements ST_{ij} represent the number of times node *i* was out of stock due to a disruption at an upstream node *j*, and let

$$DL = \begin{bmatrix} DL_{11} & . & . & DL_{1n} \\ . & & & . \\ . & & DL_{ij} & . \\ DL_{n1} & . & . & DL_{nn} \end{bmatrix}$$

be a matrix whose elements DL_{ij} represent the total number of periods node *i* was out of stock due to a disruption at an upstream node *j*. We also define the *disruption recovery* state at node *j* at time *t* as

$$DR_{ij,t} = \begin{cases} 1 & if\ DS_{ij,t} = 0 \quad and \quad DS_{ij,t-1} = 1 \\ 0 & if\ DS_{ij,t} = 0 \quad and \quad DS_{ij,t-1} = 0 \end{cases}$$

Finally, let $I_{j,t}$ be the inventory on hand at node *j* in period *t*.

4.4.1 Timed Reachability Analysis Algorithm

TRA estimates the extent of the downstream delay caused by an upstream disruption in the supply chain and determines the degree of impact on the upstream node by each of the downstream nodes. The information from the timed reachability analysis could be used to make informed adjustments to our analytic computations of safety stocks.

Our algorithm for TRA is as follows:

For $i = 2$ to n

For $k = 1$ To $k = Pcount_{im}$

For $j = 1$ To $j = |P_{im}^k|$, where $|P_{im}^k|$ is the number of elements on path P_{im}^k

% Starting with the fist node, first path, and the first node on the path cycle through

% all nodes, all paths for each node, and all nodes on each path

If $DS_{ij,t} = 1$ *Or* $DR_{ij,t} = 1$ *Then*

If $I_{j,t} = 0$ *Then*

$DS_{ij+1,t} = 1$

% Disruption state is in effect at the next down stream node when there is a disruption

% state or disruption recovery state at the node upstream and when $I_{j,t} = 0$ at the

% upstream node

If $DS_{ij,t-1} = 0$ *and* $DS_{ij,t} = 1$ *Then*

$ST_{ij} = ST_{ij} + 1$

% Increment ST_{ij} by 1 only one time for each node

End If

$DL_{ij} = DL_{ij} + 1$

% Disruption length is incremented each time there is a disruption at the node or any

% upstream nodes

End If

End If

If $j = 1$ *Then*

If $r_{i,t} = 0$ *Then*

$DS_{ii,t} = 0$

$DR_{ii,t} = 1$

$dc_{ii} = time + L_{ij}$

End If

% If a node is the node where disruption occurred then $DS_{ii,t} = 0$ when $r_{i,t} = 0$, and

% disruption recovery state begins ($DR_{ii,t} = 1$)

If $time = dc_{ii}$ *Then*

$DR_{ii,t} = 0$

$dc_{ii} = 0$

End If

End If

% $DR_{ii,t} = 0$ when the regular fulfillment time elapses

If $j > 1$ *Then*

If $DS_{ij,t} = 1$ *And* $I_{j,t} > 0$ *Then*

$DS_{ij,t} = 0$

$DR_{ij,t} = 1$

$dc_{ij} = time + L_{jj+1}$

End If

% For all other nodes $DS_{ij,t} = 0$ when inventory $I_{j,t} > 0$

If $time = dc_{ij}$ *Then*

$DR_{ij,t} = 0$

$dc_{ij} = 0$

End If

% $DR_{ii,t} = 0$ when the regular fulfillment time elapses

End If

Next

Next

Next

4.5 Informed Safety Stock Adjustment Algorithm (Adjusted Modified Graves and Willems model)

The information of interest from the timed reachability analysis is the impact of a disruption at any upstream node on the final node (n_1). This is measured by the number of periods the final node's demand is not met due to a disruption at any upstream node. The impact serves as an input to a linear optimization problem, where the objective is to minimize total cost of the additional buffer inventory while also ensuring that the supply chain achieves the desired service level at the final node.

We define SL_i as the service level at node *i* (an output from the simulation) and *T* as the simulation time (i.e., the number of periods run in the simulation). Then, the LP to find the inventory adjustment levels is as follows:

$$\min \sum_{i=2}^{n} \frac{\mu_i h_i DL_{i1} x_i}{ST_{i1}} \qquad (5)$$

subject to

$$\sum_{i=2}^{n} DL_{i1} x_i = \max\left\{\sum_{i=2}^{n} DL_{i1}, (1 - SL_1)T\right\} - (1-\alpha)T \quad (6)$$

$$0 \le x_i \le 1,\ i = 1,\ldots,n$$

The solution to this LP, $x_i \quad (i = 1,...n)$, is computed in Excel Solver and is used to compute the additional inventory, $A_i = \mu_i x_i DL_{i1} / ST_{i1}$, that needs to be added to the base stock at node i ($i = 1,...n$). The base stock adjustment is performed manually.

5. Computational Experiments, Findings and Discussion

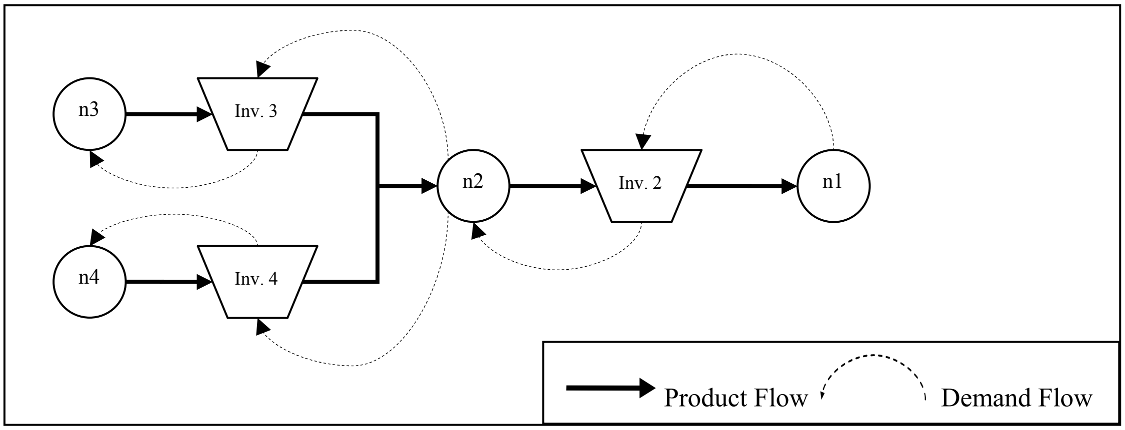
We applied and tested the methodology described above on two supply chain networks, a smaller one containing four nodes and a larger one containing eight. The smaller network was used for extensive sensitivity analysis and for deriving prescriptions for safety stock placement under different scenarios. The larger network was used to verify the scalability of our approach and to determine whether the findings from the analysis of the small network hold for the larger system.

Our experiments fall into two categories. The first is a traditional sensitivity analysis where a number of parameters are increased and/or decreased one-at-a-time by a set increment. The second is a scenario analysis where we vary either one or more factors, not in increments but once to a larger or smaller value. For each scenario, we compare three different solutions. First, the base stock levels are computed via the original Graves and Willems (2003) model (GW). Second, the base stock levels are computed via the modification to the Graves and Willems model that we discuss in Section 4 (Mod-GW). Finally, the base stock levels are adjusted when needed through the Informed Safety Stock Adjustment (ISSA) process also described in Section 4. The parameters we vary in both the sensitivity and scenario analyses are holding cost h_i, normal fulfillment time L_i, the worst case fulfillment time K_i, and probability of a disruption p_i.

5.1 Small network experiments and analysis

The four-node supply chain that is the focus of our small-network experiments is shown in Figure 2. For this supply chain network, $N = \{1,\ldots,4\}$ and $E = \{(4,2),(3,2),(2,1)\}$.

Figure 2 – Four Node Supply Chain



The base case for our experiment is shown in Table 1. We require the service level at node 1 to be 95%, meaning that $k_2 = \Phi^{-1}(0.95) = 1.645$. The service level requirements for the upstream nodes serve as a decision variables. If the system achieves the service level that falls in a 95% confidence interval around $\alpha = 0.95$, we say that the system performs up to expectations.

Table 1 – Network Parameters for the Base Case Experiments

Node	L_i	K_i	p_i	h_i	μ_j	σ_j
2	1	2	1%	10	200	30
3	3	10	1%	1	200	30
4	3	10	1%	1	200	30

5.1.1 Sensitivity Analysis Results

We are interested to know how a small change in each of the parameters affects the system performance, and when the system goes from performing below expectations to achieving the target service level. To answer these questions, we use sensitivity analysis. We change each of the parameters in small increments and simulate the system performance for each change. For the base stock calculation, we use the Mod-GW model. Figures 3 through 6 summarize our results. The black dots represent the target service level. The white squares are the simulation results. The vertical lines represent a 95% confidence interval.

Figure 3 – Service Level vs. Ratio between Normal Fulfillment and Worst Possible Fulfillment Times at node 3

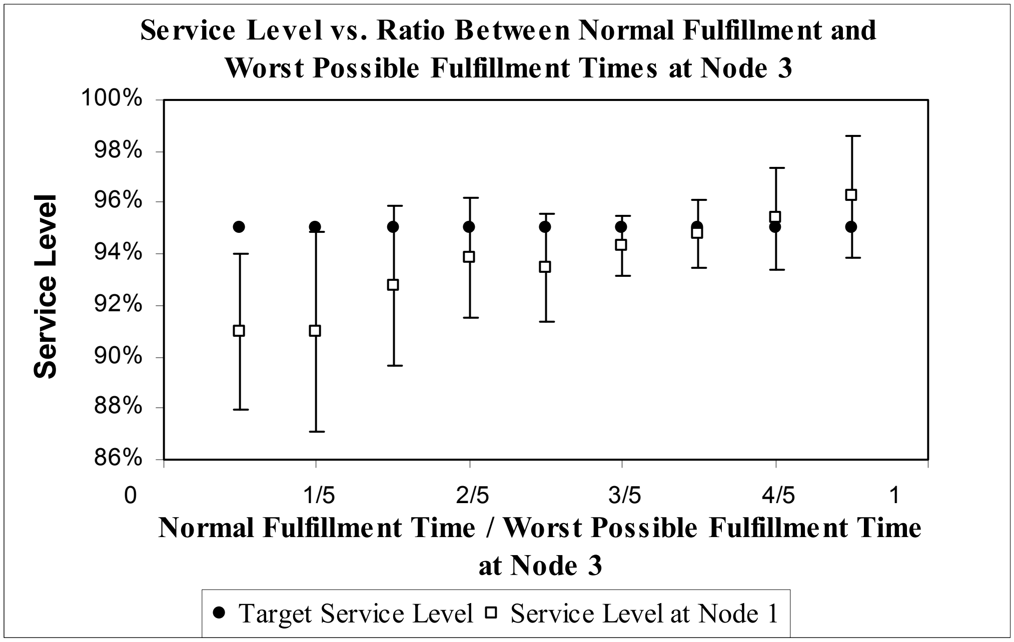


Figure 4 – Service Level vs. Ratio between Annual Holding Cost at Node 3 and Node 2

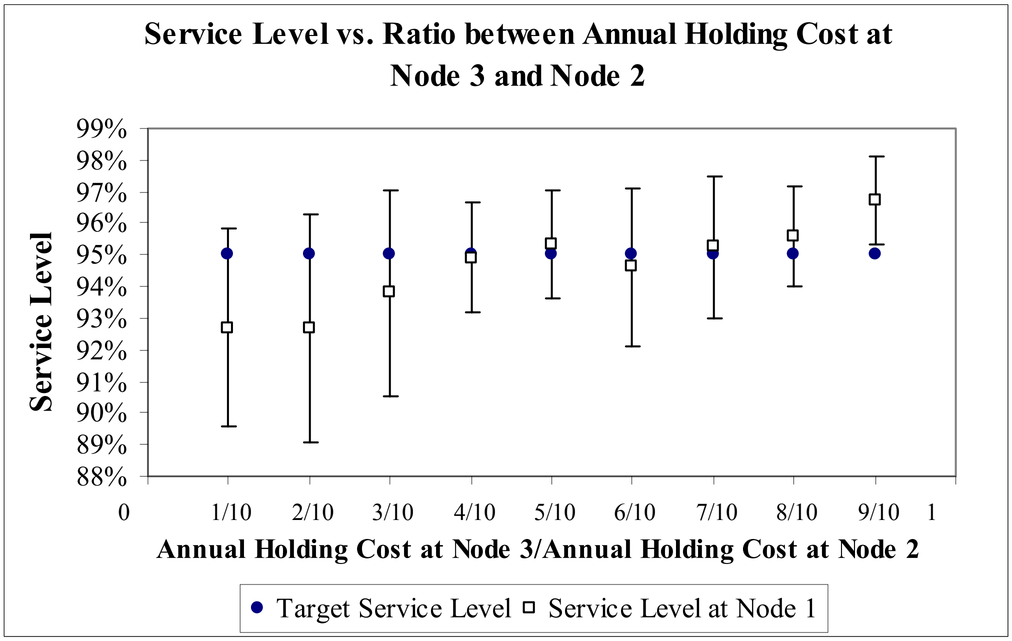


Figure 5 – Probability of Disruption at Node 3 vs. Service Level

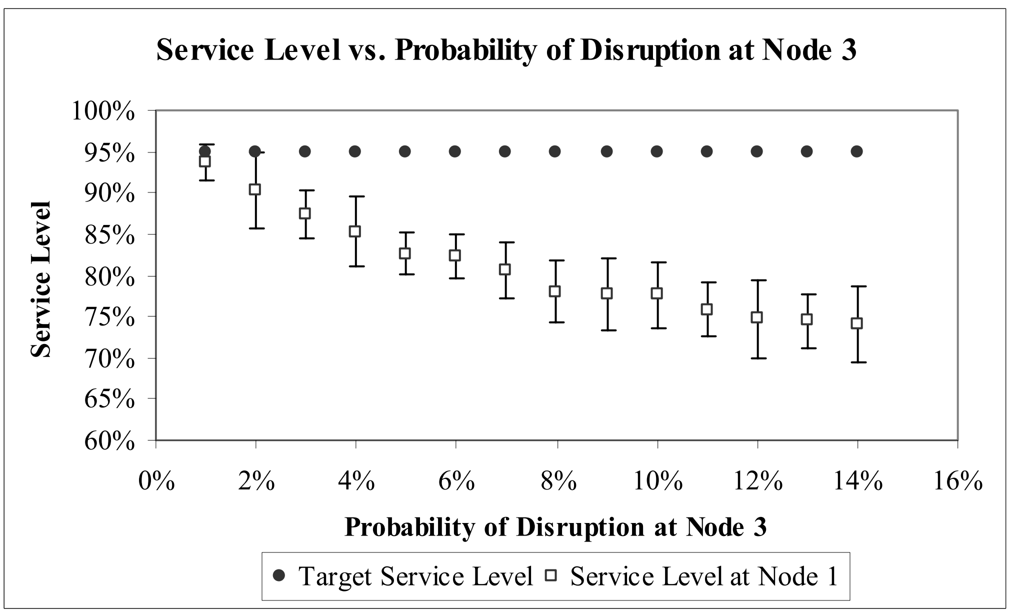
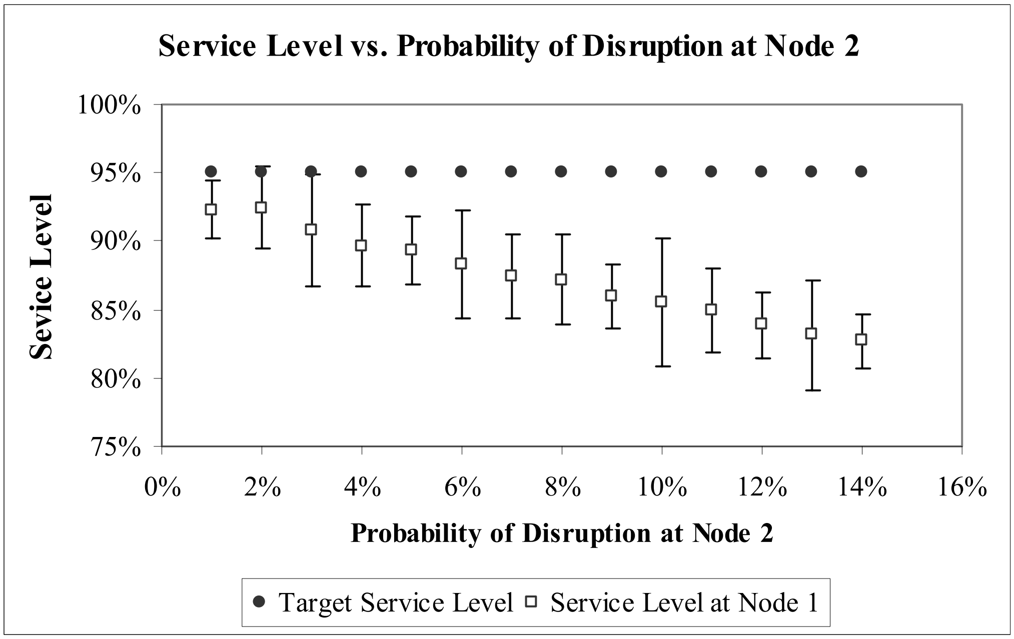


Figure 6 – Probability of Disruption at Node 2 vs. Service Level



The naïve interpretation of Figure 3 is that as the normal fulfillment time increases, the system performs better. However, remember that when we vary normal fulfillment time our worst case fulfillment time stays the same. Therefore, the results from Figure 3 should be interpreted as follows: As the gap between the normal fulfillment and worst fulfillment times shrinks, the system performs better. The same analysis can be made for the results presented in Figure 4. As we increase annual holding cost for node 3, the holding cost at node 2 remains the same. Therefore, as the costs difference between nodes 2 and 3 shrinks, the Modified Graves and Willems model suggests more safety stock being placed downstream at node 2. Also notice that the simulation results display a relatively large standard error; longer simulation runs may be necessary to allow us to achieve more stable system performance. (For more detailed data the reader is referred to Appendix I.)

5.1.2 Scenario Analysis Results

Similar to the sensitivity analysis, we compare the GW, Mod-GW, and ISSA solutions, but in this case, across various combinations of changes in the base case parameter values. Table 2 below summarizes the results obtained for the base case.

Table 2 - Base Case Results, Four-Node Supply Chain

	Base Stock Levels			Node 1 Service Level		Node 1 Fill Rate	
	Node 2	**Node 3**	**Node 4**	**Mean**	**Std. Dev.**	**Mean**	**Std. Dev.**
GW	296	694	694	91.50%	2.09%	93.10%	2.00%
Mod-GW	299	709	709	92.90%	1.80%	94.30%	1.80%
ISSA	299	1233	709	94.50%	2.19%	95.80%	2.00%

By observing the results from the base case, it is easy to notice that ISSA algorithm has suggested placing additional safety stock only at node 3. Recall that there is no difference between node 3 and 4 from the parameters stand point; thus, we believe the linear programming problem used in ISSA could have alternate optimal solutions, and we suspect that the system would perform equally well if the additional safety stock for node 3 were spread equally between nodes 3 and 4. Table 3 summarizes these results and confirms our hypothesis.

Table 3 – Base Case Results, Four-Node Supply Chain (additional Safety Stock for node 3 is split evenly between 3 and 4)

	Base Stock Levels			Node 1 Service Level		Node 1 Fill Rate	
	Node 2	**Node 3**	**Node 4**	**Mean**	**Std. Dev.**	**Mean**	**Std. Dev.**
Mod-GW	299	709	709	92.5%	2.5%	93.7%	2.1%
ISSA	299	1233	709	94.5%	2.19%	95.8%	2.0%
ISSA Manual	299	971	971	94.0%	2.1%	95.5%	1.7%

We now analyze the system performance as various problem parameters are changed one-at-a-time, and then in conjunction. Table 4 below summarizes the results for scenarios where the one of the parameters was changed from the base case value. (More extensive tables of the results can be found in Appendix I.)

Table 4 - Additional Scenario Results

Scenario		Base Stock Levels			Node 1 Service Level		Node 1 Fill Rate	
		Node 2	Node 3	Node 4	Mean	Std. Dev.	Mean	Std. Dev.
2a $(L_3 = 6)$	**GW**	304	1345	693	93.8%	1.20%	95.4%	1.1%
	Mod-GW	307	1354	708	94.6%	2.1%	95.9%	2.0%
	ISSA	307	1717	708	95.8%	2.61%	96.8%	2.5%
2b $(h_4 = 5)$	**GW**	549	655	613	94.9%	2.81%	95.7%	2.6%
	Mod-GW	556	670	628	95.4%	1.9%	96.3%	1.7%
	ISSA	556	670	628	95.4%	1.9%	96.3%	1.7%
2c $(p_3 = 0.10)$	**GW**	296	694	694	71.1%	4.91%	75.4%	4.3%
	Mod-GW	301	847	709	77.5%	5.4%	81.3%	4.8%
	ISSA	301	1488	800	91.6%	2.83%	93.8%	2.3%
2d $(p_2 = 0.10)$	**GW**	296	694	694	85.8%	1.93%	90.5%	1.7%
	Mod-GW	321	708	708	85.3%	3.4%	91.0%	2.9%
	ISSA	464	1406	1422	98.5%	1.12%	99.2%	0.7%

Further, we looked at how the experimental parameters interact with each other by changing them in conjunction with each other. Table 5 summarizes our findings. More extensive tables of the results can be found in Appendix I.

Table 5 – Scenario Analysis $(L_j \quad and \quad h_j)$

Scenario		Base Stock Levels			Node 1 Service Level		Node 1 Fill Rate	
		Node 2	Node 3	Node 4	Mean	Std. Dev.	Mean	Std. Dev.
3a $(L_3 = L_4 = 6)$	**GW**	313	1345	1345	95.60%	1.99%	97.00%	1.50%
	Mod-GW	315	1353	1353	95.10%	1.50%	96.70%	1.30%
	ISSA	315	1353	1367	96.20%	1.70%	97.50%	1.20%
3b $(L_3 = 6; h_4 = 5)$	**GW**	558	1307	614	96.90%	1.21%	97.50%	1.00%
	Mod-GW	564	1316	628	96.50%	1.60%	97.30%	1.30%
	ISSA	564	1316	628	96.50%	1.60%	97.30%	1.30%
3c $(L_3 = 6)$ $(h_3 = h_4 = 5)$	**GW**	875	1205	600	98.40%	0.86%	98.80%	1.80%
	Mod-GW	887	1213	614	98.60%	1.00%	98.90%	0.90%
	ISSA	887	1213	614	98.60%	1.00%	98.90%	0.70%
3d $(L_3 = L_4 = 6)$ $(h_3 = h_4 = 5)$	**GW**	1110	1200	1200	99.90%	0.15%	100.00%	0.00%
	Mod-GW	1118	1208	1208	100.00%	0.10%	100.00%	0.10%
	ISSA	1118	1208	1208	100.00%	0.10%	100.00%	0.10%
3e $(h_3 = h_4 = 5)$	**GW**	685	600	600	96.20%	1.97%	96.90%	1.80%
	Mod-GW	698	614	614	96.70%	1.00%	97.20%	0.90%
	ISSA	698	614	614	96.70%	1.00%	97.20%	0.90%

Table 6 – Scenario Analysis $\left(h_j, \quad p_j, \quad and \quad K_j\right)$

Scenario		Base Stock Levels			Node 1 Service Level		Node 1 Fill Rate	
		Node 2	Node 3	Node 4	Mean	Std. Dev.	Mean	Std. Dev.
4a $(h_4 = 5; p_3 = 0.10)$	**GW**	549	655	613	80.20%	3.21%	83.60%	2.70%
	Mod-GW	554	810	629	84.30%	4.40%	87.30%	3.90%
	ISSA	554	1296	629	94.40%	2.40%	95.90%	1.80%
4b $(L_3 = 6)$ $(h_4 = 5; p_3 = 0.10)$	**GW**	558	1307	614	90.10%	2.88%	92.90%	2.60%
	Mod-GW	567	1392	628	94.10%	1.90%	95.90%	1.50%
	ISSA	567	1565	628	96.60%	2.20%	97.60%	1.90%
4c $(p_2 = 0.10; K_2 = 4)$	**GW**	296	694	694	77.80%	3.10%	82.20%	2.60%
	Mod-GW	370	705	705	80.90%	3.30%	86.10%	2.50%
	ISSA	656	1370	1440	97.00%	1.00%	98.30%	0.60%
4d $(p_2 = 0.10; h_2 = 15)$	**GW**	277	706	706	83.90%	2.10%	88.40%	2.20%
	Mod-GW	301	720	720	84.20%	2.70%	89.20%	2.60%
	ISSA	443	1518	1496	97.70%	1.10%	99.50%	0.50%
4e $(p_2 = 0.10)$ $(h_2 = 15; K_2 = 4)$	**GW**	277	706	706	77.20%	3.60%	81.10%	3.20%
	Mod-GW	348	718	718	79.30%	4.10%	84.80%	3.20%
	ISSA	651	1374	1457	96.30%	1.00%	98.00%	0.70%

From scenarios 3a through 3e, we conclude that the system performs well under Mod-GW as the ratio of the disrupted fulfillment time to the normal fulfillment time (K_j / L_j) decreases (i.e., as L_j / K_j approaches 1 from below, per Figure 3) and/or as the upstream holding cost increases relative to the downstream holding cost. As the probability of disruption increases at any node, however, the system performs well below the expectation level, and the ISSA is needed to adjust for that. In certain cases such as in Scenarios 4a through 4e, the ISSA suggests too much of safety stock and forces the system to perform above the desired service level. We think the solution for ISSA can be more accurate if the simulation either covers more periods or we take more samples.

5.1.3 Discussion

Considering the results from Scenarios 2a through 2d for the Mod-GW solutions, we offer the following observations:

1. When the ratio K_j / L_j decreases by 50%, the system service level improves by 0.3 percentage points.

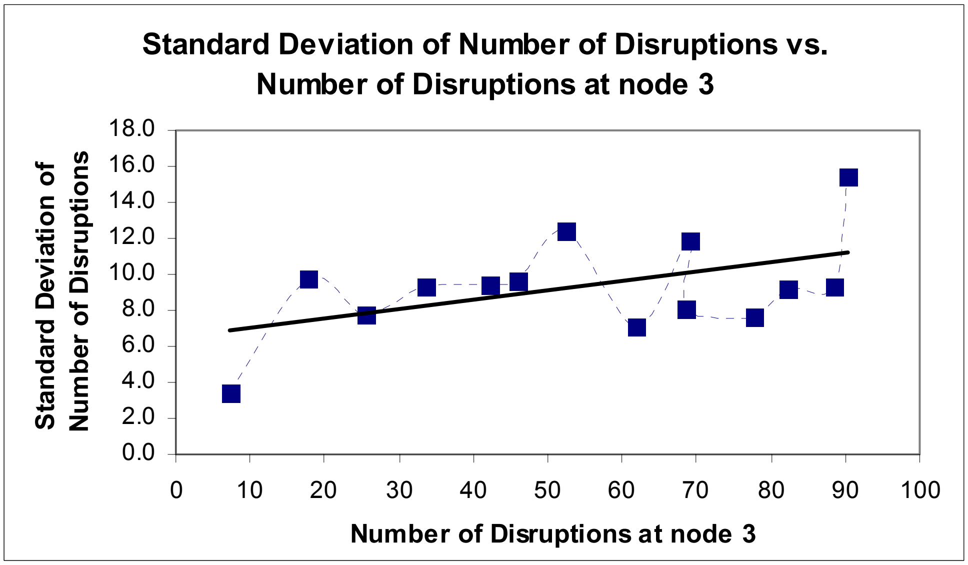
2. When the ratio of node 4 holding cost to node holding cost 2 decreases by 50%, the service level improves by 1.1 percentage points.
3. When the probability of disruption at node 3 increases 10 times, the service level drops by 16.8 percentage points.
4. When the probability of disruption at node 2 increases 10 times, the service level drops by 9 percentage points.

Let us take observation 1 and look at the sensitivity results outlined in Section 5.1.1 Sensitivity Analysis Results. As K_j / L_j gets smaller, the system performs better. We conclude that the performance of the Mod-GW model degrades as the disruption length increases relative to the normal fulfillment time. In those cases, we recommend use of the Informed Adjustment of Safety Stock Algorithm to determine an additional amount of safety stock and its placement in order to achieve the service level requirements.

We observe a similar result in respect to relative holding costs between node 4 and node 2. As h_4 decreases relative to h_2 the Mod–GW model does not suggest enough safety stock. The reason for that is, smaller the h_4 causes more incentive for the Mod–GW model to place safety stock at node 4 and not enough at 2. When we apply ISSA, it suggests additional inventory at the node that has the lower holding cost, and the resulting performance of the system comes up to the expected level. One might suspect that the ISSA should also place additional inventory at node 2. However, placing more of safety stock at node 4 significantly mitigates the possibility of a disruption at node 4 propagating to node 2.

Scenarios 2c and 2d focus on the probability of disruption at nodes 3 and 2 respectively. We first observe that increasing p_3 has a greater impact on the service level at node 1 than increasing p_2. Recall that the service level at node 1 in scenario 2c ($p_3 = 0.10$) is 77.5% and in 2d ($p_2 = 0.10$) is 85.3%. What is also important to note is that ISSA does not suggest enough of additional safety stock for scenario 2c and suggests too much for scenario 2d. We suspect that this is due to the higher variability in the number of disruptions. Figure 7 represents variability in the number of disruptions for node 2.

Figure 7 Variability in number of disruptions



From the sensitivity analysis, we observe that the probability of a disruption and the service level have an inverse, and almost linear, relationship, with a much steeper slope for changes in p_3. The slope in Figure 5 is – 1.4, and slope in

Figure 6 is – 0.69. Therefore, the probability of disruption at the supplier level appears to have a greater impact on system performance. We suspect that this is due to the fact that Mod-GW tries to place greater safety stock at the node where the chances of disruptions are higher. Therefore, as the probability of disruption increases at node 2, more safety stock at node 2 allows the system to perform better.

When looking at the scenario analysis results, it is easy to observe that if the system performs well when a particular parameter is changed, then it performs well – and even better than in the individual cases – when two or more parameters are changed in conjunction. This statement holds true for the results in Scenarios 3a through 3e, where normal fulfillment time and annual holding cost are changed. For the results in Scenarios 4a through 4e, the additional parameter varied is the probability of disruption. In those cases, the system does not perform well. Furthermore, Mod-GW performs poorly when the probability of a disruption is relatively large. Once we apply the ISSA algorithm, the system generally achieves the target service level.

We are also interested in looking at interactions between various parameters, and how they affect the system performance and safety stock placement. Consider scenarios 2c, 4a, and 4c. In scenario 2c, where the ($p_3 = 0.10$), the Mod-GW model places the bulk of the safety stock at the supplier nodes 3 and 4, but it does not adequately account for disruptions in setting these safety stock levels. The ISSA algorithm tries to increase safety stock, mainly at node 3, but also at node 4. It does not suggest any change to the base stock level at node 2. Although the performance improves, it still misses the target. When we decrease the difference between the holding costs for node 2 and node 4 (Scenario 4a), the Mod-GW solution redistributes the safety stock redistribution, with more of it going to nodes 2 and 3. As a result, the system performs better. However, the ISSA algorithm still makes additional improvements in service performance by further increasing the safety stock at node 3.

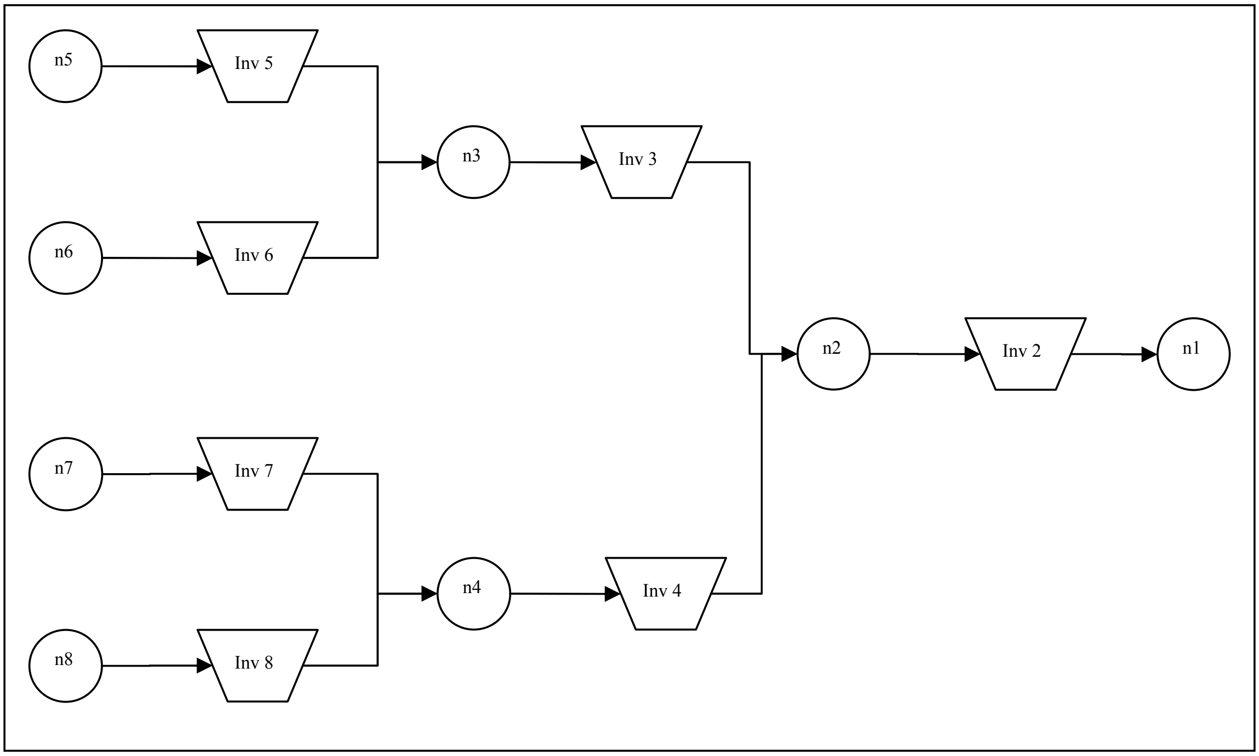
Thus, our research findings from analyzing a small supply chain network can be summarized as follows:

1. As the ratio K_j / L_j decreases, the system performs better under the Modified Graves and Willems solution
2. As the relative holding cost between an upstream node and the final stocking point decreases, the system performs better under the Modified Graves and Willems solution
3. The Modified Graves and Willems model solution does not sufficiently reflect the effects on the supply chain of an increasing probability of disruptions.
4. When the probability of disruption is increased in conjunction with holding cost or normal fulfillment time, the system performs worse under the Modified Graves and Willems solution.

5.2 Large network experiments

We are interested in testing some of our findings obtained from the extensive analysis of the small network on a larger supply chain. The larger supply chain that is the focus of our analysis is shown in Figure 8. This network has $N = \{1, 2, \ldots, 8\}$ and $E = \{(2,1),(3,2),(4,2),(5,3),(6,3),(7,4),(8,4)\}$.

Figure 8 – Eight-Node Supply Chain



The base case for our experiment is shown in Table 7. We require the service level at node 1 to be 95%, meaning that $k_2 = \Phi^{-1}(0.95) = 1.645$. The service level requirements for the upstream nodes serve as a decision variables. If the system achieves a service level that falls in 95% confidence interval around $\alpha = 0.95$, we say that the system performs up to expectations.

Table 7 – Network Parameters for the base case Experiments

Node	L_i	K_i	p_i	h_i	μ_j	σ_j
2	1	2	1%	10	200	30
3	3	10	1%	1	200	30
4	3	10	1%	1	200	30
5	3	10	1%	0.1	200	30
6	3	10	1%	0.1	200	30
7	3	10	1%	0.1	200	30
8	3	10	1%	0.1	200	30

5.2.1 Scenario Analysis

We start by first obtaining the results for the base case in three ways. First, the base stock was computed via original Graves and Willems model. Second, the base stock was computed via the Mod-GW model. Finally, the base stock was adjusted when needed through the Informed Safety Stock Adjustment process. Table 8 summarizes results obtained for the base case.

Table 8 – Base Case Results, Eight-Node Supply Chain

	Base Stock Levels							Node 1 Service Level		Node 1 Fill Rate	
	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8	Mean	Std. Dev.	Mean	Std. Dev.
GW	300	770	770	679	679	679	679	84.20%	4.70%	87%	4.10%
Mod-GW	303	787	787	694	694	694	694	84.60%	3.10%	87%	2.60%
ISSA	303	1292	787	1421	1378	1345	1405	93.60%	1.80%	95%	1.70%

Next, we would like to test if the conclusions 1 through 3 from the discussion in section 5.1.3 Discussion hold. To accomplish that, we performed scenario analysis on the large network. We looked at three scenarios where we changed the holding cost, normal fulfillment time, and probability of disruption at the first tier suppliers. Table 9 summarizes our results:

Table 9 – Additional Results for 8-Node Supply Chain

		Base Stock Levels							Node 1 Service Level		Node 1 Fill Rate	
Scenario		**Node 2**	**Node 3**	**Node 4**	**Node 5**	**Node 6**	**Node 7**	**Node 8**	**Mean**	**Std. Dev.**	**Mean**	**Std. Dev.**
5a $(h_4 = 5)$	**GW**	624	917	662	628	628	683	683	90.10%	3.00%	92%	2.20%
	Mod-GW	635	936	678	642	642	698	698	92.10%	2.30%	94%	1.80%
	ISSA	635	936	678	1169	819	1229	1218	95.30%	1.80%	97%	1.60%
5b $(p_3 = 0.10)$	**GW**	300	770	770	679	679	679	679	67.20%	5.70%	72%	5.40%
	Mod-GW	305	931	787	692	692	694	694	73.00%	5.00%	77%	4.70%
	ISSA	305	1606	908	1391	1474	1476	1419	93.80%	2.60%	96%	2.00%
5c $(L_3 = 6)$	**GW**	308	1448	770	670	670	679	679	86.00%	3.50%	89%	3.10%
	Mod-GW	311	1458	787	685	685	694	694	87.10%	3.70%	90%	3.30%
	ISSA	311	1458	1040	1290	1309	1355	1342	95.00%	2.20%	97%	1.90%

Observing the results obtained from the larger network, we conclude that our expectations hold. The chain performs better than the base case when the ratio K_j / L_j decreases, and as the relative holding cost between an upstream node and the final stocking point decreases. The ISSA algorithm suggests a sufficient amount of additional safety stock to achieve the target service level. Finally, increasing the probability of disruption at a supplier forces a system service level performance below that of the base case under the Mod-GW model, as expected. In this case, the

Informed Safety Stock Adjustment Algorithm suggests also suggests a sufficient amount of additional safety stock to achieve the target service level.

We also notice interesting safety stock re-adjustments as the system's parameters vary. Consider the base case results Table 8 and scenario 5a Table 8 results. As h_4 increases more safety stock is pushed to node 2, the base-stock for node 2 more than doubles. Notice that because node 3 is relatively cheaper more safety stock placed their. Safety stock is also rebalanced between the second tier suppliers nodes 5 – 8 although all of them identical. That forces the system to perform almost to up to expectations; however, the ISSA algorithm is stilled required. The ISSA algorithm assigns additional safety stock to nodes 5 – 8. It is interesting that nodes 7 and 8 get almost identical additional safety stock supplies, but nodes 5 and 6 are not.

It is easy to observe that as a supply chain gets larger and more complex there is more interactions between its components. It is also harder to anticipate what the best safety stock allocation is.

6. Summary of Findings

We categorize our findings into observational results and guiding managerial principles. Our observational results can be summarized as follows:

1. When upstream nodes have identical parameters and there is a need for the ISSA algorithm, the LP to allocate additional buffer stock may have multiple optimal solutions.
2. Increasing visibility to the larger supply chain by accounting for an additional echelon results in a more realistic and more accurate system performance. The resulting performance is contingent on system parameters and their interaction with each other, as above.

Although we have some concrete conclusions and observations that come from our research, we cannot provide a precise recipe to management regarding how to place safety stock in a supply chain. In general, safety stock placement depends substantially on the supply chain cost and lead-time parameters. We can, however, provide three general principles regarding important relationships in the problem parameters, as follows:

1. The relative difference in holding cost between the supplier components and finished goods has a significant impact on where safety stock is placed throughout the supply chain. If the difference is small, then more safety stock is going to be placed downstream.
2. Both the relative frequency of disruptions at upstream nodes and their length relative to the normal lead time have an impact on safety stock levels and placement in the chain. An analytic solution to compute the base stock levels in the chain does not adequately account for disruptions, particularly as their likelihood increases. Therefore, there is a need for tools that can assess the network performance and suggest the appropriate adjustments to safety stock levels and placement.
3. As the network grows larger and more complex, it becomes harder to compute analytic solutions that ensure performance up to expectations. Therefore, there is a need for heuristics to complement analytic solutions.

Summarizing our observational results and general principles, it is clear that relying solely on analytic solutions cannot guarantee the desired service performance results in a supply chain that is subject to disruptions in supply. Our model provides a way to balance the analytic and heuristic solutions so that they complement each other.

7. Recommendations for future research

The research presented in this thesis can lead to a number of follow-on studies. Three that we feel to be of particular interest are as follows:

Demand Decay: The "stochastic-service" model of Graves and Willems (2003) that serves as the basis for our research assumes that demand is fully backlogged. It would be interesting to relax that assumption and instead use a demand loss model similar to the one suggested in Warsing et al. (2000). In this case, the fraction of original demand remaining if it is fulfilled s periods late is given by

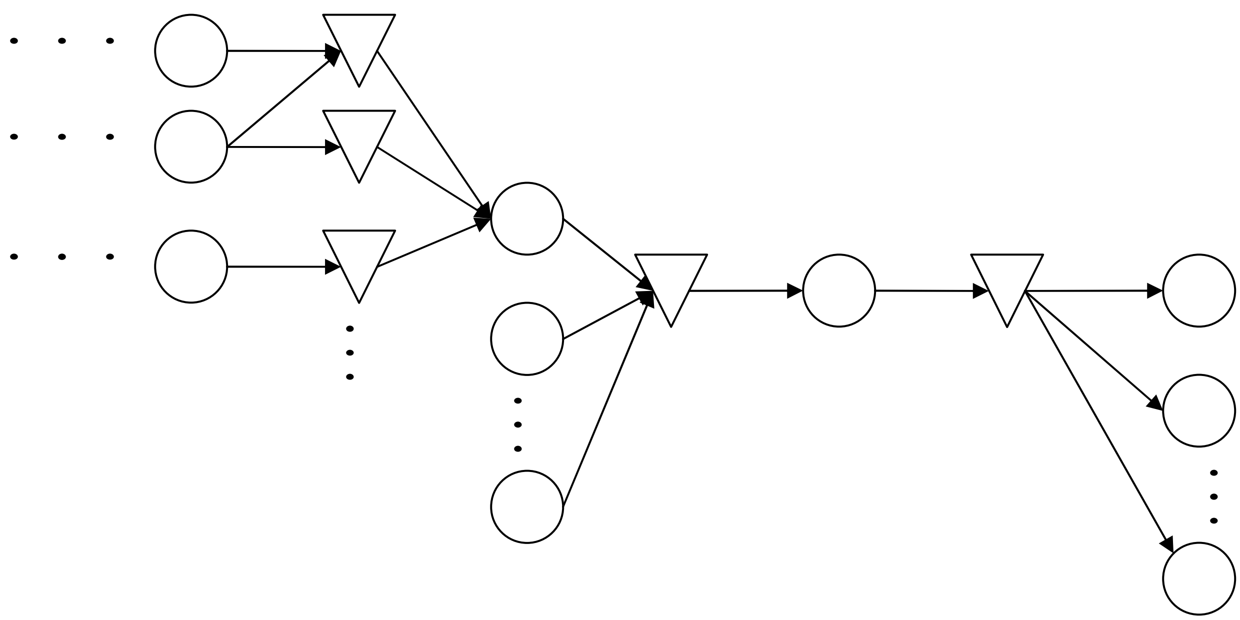
$$\delta(s) = \begin{cases} 1 - e^{-\gamma(\rho - s)} & \text{if } 0 < s < \rho \\ 0 & \text{otherwise,} \end{cases}$$

where ρ is the maximum number of periods that any amount of demand remains and γ is a scaling parameter. The importance of this model is that it allows us to study the ability of safety stock-based buffering to limit the loss of market share in the event of a supply disruption, a clearer basis upon which to justify inventory buffering than just shifting inventory costs around the supply network. Moreover, the model can be used to assess the impact of a particular buffering strategy in scenarios with a larger or smaller proportion of customers that are willing to wait for backlogged demand to be fulfilled. Since this effect would be quite difficult to account for analytically in an optimization model, the ISSA algorithm developed in this thesis is particularly well-suited to the analysis.

Continuous distribution for disruption probability: Our research assumes a discrete distribution to simulate disruptions. We are interested to know if any of our results change or if the Mod-GW model is more accurate using a continuous distribution, such as a Beta distribution.

Supply chain with divergent elements: The supply chains we study in this thesis are convergent supply chains. Additional research needs to be done to observe how the proposed methodology performs when the supply chain network has both divergent and convergent elements. An example of such a supply chain is given in Figure 9.

Figure 9 – Supply Chain with divergent elements



8. References

Anonymous, (2003). *7 Tips that help hedge against future supply chain disruptions*, Mar 2003 Supplier Selection & Management Report; 03, 3; ABI/INFORM Trade & Industry, pg. 5

Bhaskaran K., Leung, Y.T. (1997). *Manufacturing supply chain modeling and reengineering*, Sadhana—Academy Proceedings in Engineering Sciences 22 (2) 165–187.

Blackhurst, J., Wu, T. and O'Grady, P. (2004), *Network based approach to modeling uncertainty in a supply chain*, International Journal of Production Research, 42 (8), 1639-1658.

Blackhurst, J. (2005), personal communication.

Billington, C. and Lee, H. L. (1993), *Material management in decentralized supply chains*, Operations Research, 41, 835-847.

Chopra, S. and Sodhi, M. S. (2004), "*Managing Risk To Avoid Supply-Chain Breakdown*", MIT Sloan Management Review, vol. 46 iss. 1.

Chu P.C., (1997) *Actors and scripts: Object-oriented simulation of enterprise systems*, Journal of Systems Software 37 (3), 187–199.

Cohen M.A., Lee, H.L. (1998) *Strategic analysis of integrated production distribution systems: Models and methods*, Operations Research 36 (2) 216–228.

Cohen M.A., Lee, H.L. (1989). *Resource deployment analysis of global manufacturing and distribution networks*, Journal of Manufacturing and Operations Management 2 81–104.

Connors D., An, C., Buckley, S., Feigin, G., Levas, A., Nayak, N., Petrakian, R., Srinivasan, R. (1996). *Dynamic modeling of re-engineering supply chains*, Technical report, IBM Research Division.

Ettl, M., Feigin, G. E., Lin, G. Y. and Yao, D. D. (2000), *A supply network model with base-stock control and service requirements*, Operations Research, 48(2), 216-232.

Feigin G., An, C., Connors, D., Crawford, I. (1996). *Shape up, ship out*, OR/MS Today, pp. 24–30.

Geoffrion A.M., Graves, G.W. (1984). *Multi-commodity distribution system design by Bender's decomposition*, Management Science 20, 822–824.

Graves, S. and Willems, S. (2003), *Supply chain design: Safety stock placement and supply chain configuration*, Handbooks in OR & MS, Vol. 11, A. de Kok and S. Graves, eds., Elsevier.

Hicks, D.A. (1997). *A manager's guide to supply chain and logistics problem solving tools and techniques, Part I: Understanding the techniques*, IIE Solutions.

Hicks D.A., (1997). *A manager's guide to supply chain and logistics problem solving tools and techniques, Part II: Tools, companies and industries*, IIE Solutions 45 (October).

Huang E., Cheng, F.T., Yang, H.C. (1999) *Development of a collaborative and event driven supply chain information system using mobile object technology*, International Conference on Robotics and Automation, pp. 1776–1781.

Kelton, W. D., Sadowski, R. P. (2003). *Simulation with Areana 7.0*, McGraw-Hill Science/Engineering/Math; 3rd edition

Malone T.W., Smith, S.A. (1988). *Modeling the performance of organizational structures*, Operations Research, 36 (3) 421–436.

Malone T.W. (1987). *Modeling coordination in organization and markets*, Management Science 33 (3), 1317–1331.

Mujtaba M.S. (December, 1994) *Enterprise modeling and simulation: Complex dynamic behavior of a simple model of manufacturing*, Hewlett-Packard Journal, 45, 80–114.

Nissen M.E. (2000). Supply chain process and agent design for ecommerce, http://web.nps.navy.mil/menissen.

Parker, J. (Sep 25, 2000). *Lean, mean, vulnerable*. Traffic World, 263, 13; ABI/INFORM Global pg. 25

Petri C. (1962), *Kommunikation mit Autimaten*. Ph.D. Dissertation, University of Bonn, Bonn, West Germany.

Raghavan N.R.S. (November, 1998). *Performance Analysis and Scheduling of Manufacturing Supply Chain Networks*, PhD thesis, Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India.

Swaminathan J.M., Smith, S.F., Sadeh, N.M. (1997). *A multi-agent framework for modeling supply chain dynamics*, Decision Sciences 29, 189–208.

Tayur S.R., Ganeshan, M. (1999). *Quantative Models for Supply Chain Management*, Kluwer Academic Publishers.

Viswanadham N., Narahari, Y. (1992). *Performance Modeling of Automated Manufacturing Systems*, Prentice Hall, Englewood Cliffs, NJ.

Viswanadham N., Raghavan, N.R.S. (December, 1998). *Petri net models for performance analysis of supply chain networks*, Operations Management Workshop, Indian Institute of Management, Bangalore.

Vollman T.E., Berry, W.L., Whybark, D.C. (1997). *Manufacturing Planning and Control Systems*, The Dow Jones-Irwin/APICS Series in Production Management.

Warsing, D. P., Greis, N. P. and Kasarda, J. D. (2000), *An analysis of transportation mode trade-offs in a multi-region production-distribution network*, Kenan-Flagler Business School working paper, Chapel Hill, NC.

Wee H.M., Yang, P.C. (2004). *The optimal and heuristic solutions of a distribution network*, European Journal of Operational Research 158, 626–632

Appendix I

Complete Results for Four-Node Supply Chain

Base Case Four-Node Supply Chain

	Base Stock Levels Node 2	Base Stock Levels Node 3	Base Stock Levels Node 4	Additional Safety Stock Node 2	Additional Safety Stock Node 3	Additional Safety Stock Node 4	Mean Disruption Count Node 2	Mean Disruption Count Node 3	Mean Disruption Count Node 4	Std. Dev. Disruption Count Node 2	Std. Dev. Disruption Count Node 3	Std. Dev. Disruption Count Node 4
Mod-GW	299	709	709				10.7	9.6	9.2	3.8	5.0	3.3
GW	296	694	694				8.9	10.9	11.3	3.6	4.8	5.1
ISSA	299	1233	709	0	524	0	10.3	8.1	9.2	3.6	5.6	6.6

	ST_{1j} Node 2	ST_{1j} Node 3	ST_{1j} Node 4	DL_{1j} Node 2	DL_{1j} Node 3	DL_{1j} Node 4	Average Ending Inventory Node 2	Average Ending Inventory Node 3	Average Ending Inventory Node 4	Service Level Node 1 Mean	Service Level Node 1 Std. Dev.	Fill Rate Node 1 Mean	Fill Rate Node 1 Std. Dev.
Mod-GW	10.7	9.5	9	12.8	29.4	33	91	122	119	92.9%	1.8%	94.3%	1.8%
GW	8.9	10.4	10.9	13.10	40.20	36	87	108	111	91.5%	2.09%	93.1%	2.0%
ISSA	10.3	5.2	8.9	11.5	10.8	34	92	635	108	94.5%	2.19%	95.8%	2.0%

2a ($L_i = 6$)

	Base Stock Levels Node 2	Base Stock Levels Node 3	Base Stock Levels Node 4	Additional Safety Stock Node 2	Additional Safety Stock Node 3	Additional Safety Stock Node 4	Mean Disruption Count Node 2	Mean Disruption Count Node 3	Mean Disruption Count Node 4	Std. Dev. Disruption Count Node 2	Std. Dev. Disruption Count Node 3	Std. Dev. Disruption Count Node 4
	307	1354	708	0	429	675	9.9	9.1	9.6	6.1	6.4	6.5
	304	1345	693				9.2	9.7	10	2.4	6.8	6.1
	307	1717	708	0	363	0	9.4	10.6	9.2	7.6	5.9	6.0

	ST_{1j} Node 2	ST_{1j} Node 3	ST_{1j} Node 4	DL_{1j} Node 2	DL_{1j} Node 3	DL_{1j} Node 4	Average Ending Inventory Node 2	Average Ending Inventory Node 3	Average Ending Inventory Node 4	Service Level Node 1 Mean	Service Level Node 1 Std. Dev.	Fill Rate Node 1 Mean	Fill Rate Node 1 Std. Dev.
Mod-GW	9.7	7.7	8.5	12.4	16.5	29	100	167	110	94.6%	2.1%	95.9%	2.0%
GW	9.2	8.4	9.8	10.50	17.40	36	96	166	98	93.8%	1.20%	95.4%	1.1%
ISSA	9.3	2.8	8.9	10.7	3.2	30	102	529	107	95.8%	2.61%	96.8%	2.5%

2b ($h_i = 5$)

	Base Stock Levels Node 2	Base Stock Levels Node 3	Base Stock Levels Node 4	Additional Safety Stock Node 2	Additional Safety Stock Node 3	Additional Safety Stock Node 4	Mean Disruption Count Node 2	Mean Disruption Count Node 3	Mean Disruption Count Node 4	Std. Dev. Disruption Count Node 2	Std. Dev. Disruption Count Node 3	Std. Dev. Disruption Count Node 4
Mod-GW	556	670	628	0	576	651	9.3	9.5	10.1	8.5	4.9	5.2
GW	549	655	613				10.6	10.4	9.2	3.3	5.8	2.8
ISSA	556	670	628	0	0	0	9.3	9.5	10.1	8.5	4.9	5.2

	ST_{1j} Node 2	ST_{1j} Node 3	ST_{1j} Node 4	DL_{1j} Node 2	DL_{1j} Node 3	DL_{1j} Node 4	Average Ending Inventory Node 2	Average Ending Inventory Node 3	Average Ending Inventory Node 4	Service Level Node 1 Mean	Service Level Node 1 Std. Dev.	Fill Rate Node 1 Mean	Fill Rate Node 1 Std. Dev.
Mod-GW	0.9	7.6	7.8	2.9	21.9	25	324	98	54	95.4%	1.9%	96.3%	1.7%
GW	0.9	8.8	8.7	2.10	27.10	27	312	89	49	94.9%	2.81%	95.7%	2.6%
ISSA	0.9	7.6	7.8	2.9	21.9	25	324	98	54	95.4%	1.9%	96.3%	1.7%

2c ($p_i = 0.10$)

	Base Stock Levels Node 2	Base Stock Levels Node 3	Base Stock Levels Node 4	Additional Safety Stock Node 2	Additional Safety Stock Node 3	Additional Safety Stock Node 4	Mean Disruption Count Node 2	Mean Disruption Count Node 3	Mean Disruption Count Node 4	Std. Dev. Disruption Count Node 2	Std. Dev. Disruption Count Node 3	Std. Dev. Disruption Count Node 4
Mod-GW	301	847	709	29	641	826	11.8	71.1	8.6	6.1	14.6	3.5
GW	296	694	694				10.3	71.5	9.7	5.0	9.6	5.3
ISSA	301	1488	800	0	641	91	11.8	69.6	10.8	7.1	12.3	5.6

	ST_{1j} Node 2	ST_{1j} Node 3	ST_{1j} Node 4	DL_{1j} Node 2	DL_{1j} Node 3	DL_{1j} Node 4	Average Ending Inventory Node 2	Average Ending Inventory Node 3	Average Ending Inventory Node 4	Service Level Node 1 Mean	Service Level Node 1 Std. Dev.	Fill Rate Node 1 Mean	Fill Rate Node 1 Std. Dev.
Mod-GW	11.6	61.2	7.7	21.7	196	32	76	197	197	77.5%	5.4%	81.3%	4.8%
GW	9.9	66.9	8.9	17.30	258.90	39	66	82	229	71.1%	4.91%	75.4%	4.3%
ISSA	11.6	23.7	10	13.9	42.9	36	92	758	203	91.6%	2.83%	93.8%	2.3%

2d ($p_2 = 0.10$)

	Base Stock Levels Node 2	Base Stock Levels Node 3	Base Stock Levels Node 4	Additional Safety Stock Node 2	Additional Safety Stock Node 3	Additional Safety Stock Node 4	Mean Disruption Count Node 2	Mean Disruption Count Node 3	Mean Disruption Count Node 4	Std. Dev. Disruption Count Node 2	Std. Dev. Disruption Count Node 3	Std. Dev. Disruption Count Node 4
Mod-GW	321	708	708	209	698	714	95.2	9.3	9.7	12.5	6.9	6.0
GW	296	694	694				88.8	7.6	9.3	12.2	4.3	5.3
ISSA	464	1406	1422	143	698	714	88.9	11.8	10	20.5	4.6	6.2

	ST_{1j} Node 2	ST_{1j} Node 3	ST_{1j} Node 4	DL_{1j} Node 2	DL_{1j} Node 3	DL_{1j} Node 4	Average Ending Inventory Node 2	Average Ending Inventory Node 3	Average Ending Inventory Node 4	Service Level Node 1 Mean	Service Level Node 1 Std. Dev.	Fill Rate Node 1 Mean	Fill Rate Node 1 Std. Dev.
Mod-GW	90.7	9.6	9.8	114.7	33.5	35	102	123	123	85.3%	3.4%	91.0%	2.9%
GW	86.5	7.4	9	102.50	28.90	36	81	110	105	85.8%	1.93%	90.5%	1.7%
ISSA	7.5	3.7	2.3	9.9	6.2	4	242	784	804	98.5%	1.12%	99.2%	0.7%

3a $(L_3 = L_4 = 6)$

	Base Stock Levels			Additional Safety Stock			Mean Disruption Count			Std. Dev. Disruption Count		
	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4
Mod-GW	315	1353	1353				10.2	10.3	11.6	5.2	7.2	4.7
GW	313	1345	1345				10.1	9.6	9.9	3.8	4.1	6.7
ISSA	315	1353	1367	0	0	14	9	10	8	3.2	6.6	4.1

	ST_1			DL_1			Average Ending Inventory			Service Level Node 1		Fill Rate Node 1	
	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Mean	Std. Dev.	Mean	Std. Dev.
Mod-GW	10	9.1	10	10.9	19.8	20	108	157	157	95.1%	1.5%	96.7%	1.3%
GW	9.9	9	7.5	10.40	19.10	16	106	150	151	95.6%	1.99%	97.0%	1.5%
ISSA	8.6	8.2	7.1	9.5	16.1	14	110	157	173	96.2%	1.7%	97.5%	1.2%

3b $(L_3 = 6, h_4 = 5)$

	Base Stock Levels			Additional Safety Stock			Mean Disruption Count			Std. Dev. Disruption Count		
	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4
Mod-GW	564	1316	628	0	305	600	10.7	11.4	10.6	3.7	8.2	4.7
GW	558	1307	614				9.7	9.1	9.5	8.2	4.3	4.1
ISSA	564	1316	628	0	0	0	10.7	11.4	10.6	3.7	8.2	4.7

	ST_1			DL_1			Average Ending Inventory			Service Level Node 1		Fill Rate Node 1	
	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Mean	Std. Dev.	Mean	Std. Dev.
Mod-GW	0.9	5.7	9.3	1.9	8.7	28	333	146	43	96.5%	1.6%	97.3%	1.3%
GW	0.5	5.1	7.9	1.30	7.30	24	326	143	34	96.9%	1.21%	97.5%	1.0%
ISSA	0.9	5.7	9.3	1.9	8.7	28	333	146	43	96.5%	1.6%	97.3%	1.3%

3c $(L_3 = 6; h_3 = h_4 = 5)$

	Base Stock Levels			Additional Safety Stock			Mean Disruption Count			Std. Dev. Disruption Count		
	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4
Mod-GW	887	1213	614				8.7	9.3	8.5	4.8	4.5	6.2
GW	875	1205	600				9.8	8.2	10.6	7.1	4.4	6.2
ISSA	887	1213	614	0	0	0	8.7	9.3	8.5	4.8	4.5	6.2

	ST_1			DL_1			Average Ending Inventory			Service Level Node 1		Fill Rate Node 1	
	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Mean	Std. Dev.	Mean	Std. Dev.
Mod-GW	0	2	5.4	0	2.2	12	634	61	48	98.6%	1.0%	98.9%	0.7%
GW	0.3	1.8	6.4	0.90	2.10	15	614	61	38	98.4%	0.86%	98.8%	0.6%
ISSA	0	2	5.4	0	2.2	12	634	61	48	98.6%	1.0%	98.9%	0.7%

3d $(L_3 = L_4 = 6; h_3 = h_4 = 5)$

	Base Stock Levels			Additional Safety Stock			Mean Disruption Count			Std. Dev. Disruption Count		
	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4
Mod-GW	1118	1208	1208				10.5	8.6	9.6	5.1	4.9	4.8
GW	1110	1200	1200				10.1	8.9	10.2	5.3	4.8	3.5
ISSA	1118	1208	1208	0	0	0	10.5	8.6	9.6	5.1	4.9	4.8

	ST_1			DL_1			Average Ending Inventory			Service Level Node 1		Fill Rate Node 1	
	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Mean	Std. Dev.	Mean	Std. Dev.
Mod-GW	0.1	0.3	0.3	0.1	0.3	0	876	43	42	100.0%	0.1%	100.0%	0.1%
GW	0.1	0.4	0.3	0.10	0.40	0	864	40	40	99.9%	0.15%	100.0%	0.0%
ISSA	0.1	0.3	0.3	0.1	0.3	0	876	43	42	100.0%	0.1%	100.0%	0.1%

3e $(h_3 = h_4 = 5)$

	Base Stock Levels			Additional Safety Stock			Mean Disruption Count			Std. Dev. Disruption Count		
	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4
Mod-GW	698	614	614				8.6	8.5	8.9	5.1	3.6	3.6
GW	685	600	600				11.9	9.6	9.1	5.6	6.6	4.2
ISSA	698	614	614	0	0	0	8.6	8.5	8.9	5.1	3.6	3.6

	ST_1			DL_1			Average Ending Inventory			Service Level Node 1		Fill Rate Node 1	
	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Mean	Std. Dev.	Mean	Std. Dev.
Mod-GW	0.8	6.3	6.2	2.3	17.2	17	458	47	47	96.7%	1.0%	97.2%	0.9%
GW	0.2	7.3	7.6	0.40	18.00	21	435	40	38	96.2%	1.97%	96.9%	1.8%
ISSA	0.8	6.3	6.2	2.3	17.2	17	458	47	47	96.7%	1.0%	97.2%	0.9%

4a ($h_4 = 5; p_3 = 0.10$)

	Base Stock Levels			Additional Safety Stock			Mean Disruption Count			Std. Dev. Disruption Count		
	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4
Mod-GW	554	810	629				11.4	74.1	9	6.2	11.3	5.6
GW	549	655	613				12.2	70.6	8.9	4.1	10.4	4.4
ISSA	554	1296	629	0	486	0	9	71.2	10.3	5.9	10.2	3.5

	ST_1			DL_1			Average Ending Inventory			Service Level Node 1		Fill Rate Node 1	
	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Mean	Std. Dev.	Mean	Std. Dev.
Mod-GW	3.3	52.4	8.6	8.3	137.6	29	269	176	142	84.3%	4.4%	87.3%	3.9%
GW	4.2	57.9	8.5	12.80	179.10	26	247	68	167	80.2%	3.21%	83.6%	2.7%
ISSA	0.8	20.5	9.8	1.6	30.4	28	312	590	61	94.4%	2.4%	95.9%	1.8%

4b ($h_4 = 5; p_3 = 0.10, L_3 = 6$)

	Base Stock Levels			Additional Safety Stock			Mean Disruption Count			Std. Dev. Disruption Count		
	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4
Mod-GW	567	1392	628				11.9	79.7	8	5.3	8.6	4.3
GW	558	1307	614				9	82.4	10.8	3.9	10.3	6.0
ISSA	567	1565	628	0	173	0	8.4	81.5	8.9	6.2	7.7	5.6

	ST_1			DL_1			Average Ending Inventory			Service Level Node 1		Fill Rate Node 1	
	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Mean	Std. Dev.	Mean	Std. Dev.
Mod-GW	1.8	29.3	7.3	3.5	39.7	23	315	188	73	94.1%	1.9%	95.9%	1.5%
GW	2.8	47.3	11.2	5.30	73.00	34	285	123	81	90.1%	2.88%	92.9%	2.6%
ISSA	1	11.9	7.2	1.8	14.2	22	331	339	55	96.6%	2.2%	97.6%	1.9%

4c ($p_2 = 0.10; K_2 = 4$)

	Base Stock Levels			Additional Safety Stock			Mean Disruption Count			Std. Dev. Disruption Count		
	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4
Mod-GW	370	705	705				86.5	8.4	9.2	14.7	3.6	6.3
GW	296	694	694				81.5	9.4	10.4	17.3	5.3	4.3
ISSA	656	1370	1440	286	665	735	79	9.3	9.3	10.2	5.0	5.9

	ST_1			DL_1			Average Ending Inventory			Service Level Node 1		Fill Rate Node 1	
	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Mean	Std. Dev.	Mean	Std. Dev.
Mod-GW	79.7	8.3	9.2	164.1	27.6	34	133	121	115	80.9%	3.3%	86.1%	2.5%
GW	79.6	9.5	10.1	181	38.7	41	74	111	109	77.8%	3.1%	82.2%	2.6%
ISSA	25.3	2.2	1.2	27.3	2.9	2	403	753	824	97.0%	1.0%	98.3%	0.6%

4d ($p_2 = 0.10; h_2 = 15$)

	Base Stock Levels			Additional Safety Stock			Mean Disruption Count			Std. Dev. Disruption Count		
	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4
Mod-GW	301	720	720				94.5	10.4	10.2	12.4	4.8	5.2
GW	277	706	706				92.5	10.6	8.1	12.4	4.2	4.9
ISSA	443	1518	1496	142	798	776	91.3	9.6	9.8	16.8	4.9	4.6

	ST_1			DL_1			Average Ending Inventory			Service Level Node 1		Fill Rate Node 1	
	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Mean	Std. Dev.	Mean	Std. Dev.
Mod-GW	91.7	10.1	9.8	115.1	40.3	38	84	134	138	84.2%	2.7%	89.2%	2.6%
GW	90.4	10.7	7.6	114	44	34	64	118	123	83.9%	2.1%	88.4%	2.2%
ISSA	19.7	1.5	2.1	20	1.8	3	223	899	877	97.7%	1.1%	99.5%	0.5%

4e ($p_2 = 0.10; h_2 = 15; K_2 = 4$)

	Base Stock Levels			Additional Safety Stock			Mean Disruption Count			Std. Dev. Disruption Count		
	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4
Mod-GW	348	718	718				83.5	10.8	8.9	19.4	6.5	3.8
GW	277	706	706				83.8	8.6	10.5	12.2	4.4	5.3
ISSA	651	1374	1457	303	656	739	85.7	8.9	11.2	16.1	6.1	5.8

	ST_1			DL_1			Average Ending Inventory			Service Level Node 1		Fill Rate Node 1	
	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Node 2	Node 3	Node 4	Mean	Std. Dev.	Mean	Std. Dev.
Mod-GW	80.4	11.4	8.5	171.9	37.4	31	115	129	132	79.3%	4.1%	84.8%	3.2%
GW	81.4	8.7	10.6	189.7	37.9	43	59	122	120	77.2%	3.6%	81.1%	3.2%
ISSA	29.4	2	2.2	34.7	2.7	3	394	761	834	96.3%	1.0%	98.0%	0.7%

Appendix II

Complete results for Eight-Node Supply Chain

		Base Stock							Additional Safety Stock						
		Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8
Base Case Eight-Node Supply Chain	Mod-GW	303	787	787	694	694	694	694							
	GW	300	770	770	679	679	679	679							
	ISSA	303	1292	787	1421	1378	1345	1405	0	505	0	727	684	651	711
		Mean Disruption Count							**Std. Dev. Disruption Count**						
		Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8
	Mod-GW	7.5	8.8	9.0	11.9	10.4	8.8	11.1	2.8	5.7	4.1	4.8	7.0	5.7	5.9
	GW	10.0	8.8	9.1	10.1	10.4	9.5	8.8	7.8	4.1	4.2	5.9	5.5	5.6	5.8
	ISSA	9.4	10.7	10.8	8.7	12.5	9.9	10.1	5.1	5.9	4.5	5.9	3.8	7.4	5.6
		ST_{1j}							DL_{1j}						
		Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8	Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8
	Mod-GW	7.2	8.4	7.9	10.1	9.5	7.4	9.7	11	31.4	28.8	36.7	32.5	24.1	34.5
	GW	9.9	8.4	8.9	9.3	9.2	8.1	8.3	14.8	31.1	32	30.5	31.8	27.9	27.5
	ISSA	9.1	5	10	0.5	0.4	3.6	3.2	9.2	10.4	35.8	0.8	1	8.7	5.8
		Average Ending Inventory							**Service Level Node 1**		**Fill Rate Node 1**				
		Node 2	Node 3	Node 4	Node 5	Node 6	Node 7	Node 8	Mean	Std. Dev.	Mean	Std. Dev.			
	Mod-GW	85	202	206	108	113	114	105	84.6%	3.1%	87%	2.6%			
	GW	83	189	194	100	97	96	97	84.2%	4.7%	87%	4.1%			
	ISSA	96	688	180	807	754	729	785	93.6%	1.8%	95%	1.7%			

5a

$(h_4 = 5)$

	Base Stock							Additional Safety Stock						
	Node 2	**Node 3**	**Node 4**	**Node 5**	**Node 6**	**Node 7**	**Node 8**	**Node 2**	**Node 3**	**Node 4**	**Node 5**	**Node 6**	**Node 7**	**Node 8**
Mod-GW	635	936	678	642	642	698	698							
GW	624	917	662	628	628	683	683							
ISSA	635	936	678	1169	819	1229	1218	0	0	0	527	177	531	520
	Mean Disruption Count							**Std. Dev. Disruption Count**						
	Node 2	**Node 3**	**Node 4**	**Node 5**	**Node 6**	**Node 7**	**Node 8**	**Node 2**	**Node 3**	**Node 4**	**Node 5**	**Node 6**	**Node 7**	**Node 8**
Mod-GW	10.3	9.8	8.5	11.2	9.6	10.0	11.2	5.0	4.0	6.4	6.2	6.8	6.2	4.6
GW	8.8	10.4	10.5	9.8	11.5	10.2	11.1	5.3	6.2	2.9	7.6	6.9	5.5	5.4
ISSA	8.9	10.6	10.9	9.0	10.9	9.1	10.7	4.3	6.3	5.5	4.0	4.1	5.8	5.7
	ST_{1j}							DL_{1j}						
	Node 2	**Node 3**	**Node 4**	**Node 5**	**Node 6**	**Node 7**	**Node 8**	**Node 2**	**Node 3**	**Node 4**	**Node 5**	**Node 6**	**Node 7**	**Node 8**
Mod-GW	1.8	5.3	6.8	6.3	5.3	7.2	8.0	5.3	15.4	19.8	16.6	14.2	19.1	20.8
GW	0.7	7.1	9.5	5.9	8.3	8.5	9.9	1.1	17.5	28.9	15.8	21.1	23.6	28.5
ISSA	0.7	6.9	8.4	0.8	3.5	2	4	1	14.8	22.4	1.5	6.5	2.8	7.1
	Average Ending Inventory							**Service Level Node 1**		**Fill Rate Node 1**				
	Node 2	**Node 3**	**Node 4**	**Node 5**	**Node 6**	**Node 7**	**Node 8**	**Mean**	**Std. Dev.**	**Mean**	**Std. Dev.**			
Mod-GW	378	347	101	67	71	115	113	92.1%	2.3%	94%	1.8%			
GW	358	333	92	62	57	106	100	90.1%	3.0%	92%	2.2%			
ISSA	395	342	92	569	215	619	601	95.3%	1.8%	97%	1.6%			

		Base Stock							Additional Safety Stock						
5b		**Node 2**	**Node 3**	**Node 4**	**Node 5**	**Node 6**	**Node 7**	**Node 8**	**Node 2**	**Node 3**	**Node 4**	**Node 5**	**Node 6**	**Node 7**	**Node 8**
	Mod-GW	305	931	787	692	692	694	694							
$(p_3 = 0.10)$	**GW**	300	770	770	679	679	679	679							
	ISSA	305	1606	908	1391	1474	1476	1419	0	675	121	699	782	782	725
		Mean Disruption Count							**Std. Dev. Disruption Count**						
		Node 2	**Node 3**	**Node 4**	**Node 5**	**Node 6**	**Node 7**	**Node 8**	**Node 2**	**Node 3**	**Node 4**	**Node 5**	**Node 6**	**Node 7**	**Node 8**
	Mod-GW	10.0	70.2	10.3	10.4	10.9	10.1	9.4	4.7	12.8	5.6	3.5	3.0	6.0	6.4
	GW	10.6	72.1	9.8	9.3	9.1	9.8	9.0	7.1	10.9	3.3	7.5	4.4	3.2	4.0
	ISSA	9.9	70.0	9.8	9.8	8.0	10.3	10.2	5.5	8.2	6.9	5.2	5.3	6.5	5.0
		ST_{1j}							DL_{1j}						
		Node 2	**Node 3**	**Node 4**	**Node 5**	**Node 6**	**Node 7**	**Node 8**	**Node 2**	**Node 3**	**Node 4**	**Node 5**	**Node 6**	**Node 7**	**Node 8**
	Mod-GW	10.0	57.0	9.4	10.3	11.1	8.9	7.7	17.4	192.4	38.3	36	43.4	34.8	27.9
	GW	10.6	65.9	8.8	11.4	10	8.2	7.6	23	254.2	35.5	44.8	46.4	33.3	31
	ISSA	9.8	17.3	8.6	1	0.5	1	1.9	10.9	25.6	26.6	1.8	0.9	1.5	3.4
		Average Ending Inventory							**Service Level Node 1**		**Fill Rate Node 1**				
		Node 2	**Node 3**	**Node 4**	**Node 5**	**Node 6**	**Node 7**	**Node 8**	**Mean**	**Std. Dev.**	**Mean**	**Std. Dev.**			
	Mod-GW	74	267	263	112	108	109	111	73.0%	5.0%	77%	4.7%			
	GW	65	145	293	94	95	95	96	67.2%	5.7%	72%	5.4%			
	ISSA	97	854	302	773	859	859	799	93.8%	2.6%	96%	2.0%			

		Base Stock							Additional Safety Stock						
5c		**Node 2**	**Node 3**	**Node 4**	**Node 5**	**Node 6**	**Node 7**	**Node 8**	**Node 2**	**Node 3**	**Node 4**	**Node 5**	**Node 6**	**Node 7**	**Node 8**
$(L_3 = 6)$	**Mod-GW**	311	1458	787	685	685	694	694							
	GW	308	1448	770	670	670	679	679							
	ISSA	311	1458	1040	1290	1309	1355	1342	0	0	253	605	624	661	648
		Mean Disruption Count							**Std. Dev. Disruption Count**						
		Node 2	**Node 3**	**Node 4**	**Node 5**	**Node 6**	**Node 7**	**Node 8**	**Node 2**	**Node 3**	**Node 4**	**Node 5**	**Node 6**	**Node 7**	**Node 8**
	Mod-GW	9.5	9.3	9.1	9.7	9.0	8.2	9.9	3.3	5.0	4.6	5.3	7.1	4.7	5.6
	GW	10.0	10.0	7.6	9.8	9.7	10.5	10.1	6.9	6.9	2.3	4.2	3.9	4.0	7.1
	ISSA	9.9	9.9	9.8	11.1	10.4	9.8	10.8	4.7	4.1	6.5	5.7	4.8	4.8	4.6
		ST_{1j}							DL_{1j}						
		Node 2	**Node 3**	**Node 4**	**Node 5**	**Node 6**	**Node 7**	**Node 8**	**Node 2**	**Node 3**	**Node 4**	**Node 5**	**Node 6**	**Node 7**	**Node 8**
	Mod-GW	9.5	7.7	8.6	8.7	7.6	6.9	9.1	13.5	17.5	29.9	26.3	23.7	22.8	29.5
	GW	9.9	8	7.1	8.6	8.6	10.3	8.7	14.8	19.8	25.7	29.5	28.6	34.6	30.4
	ISSA	9.5	7.2	6.3	3.6	3.3	1.5	1.3	10.4	12.1	19.8	5.4	5.9	2.6	2.4
		Average Ending Inventory							**Service Level Node 1**		**Fill Rate Node 1**				
		Node 2	**Node 3**	**Node 4**	**Node 5**	**Node 6**	**Node 7**	**Node 8**	**Mean**	**Std. Dev.**	**Mean**	**Std. Dev.**			
	Mod-GW	95	274	193	101	103	110	106	87.1%	3.7%	90%	3.3%			
	GW	91	264	179	88	90	96	97	86.0%	3.5%	89%	3.1%			
	ISSA	105	264	428	676	695	739	727	95.0%	2.2%	97%	1.9%			

Appendix III

VBA for Four-Node Supply Chain Simulation

```
Sub ThreeNodeSC()
Dim inventory(1 To 4) As Double
Dim D(1 To 4) As Double
Dim unsat(0 To 5, 0 To 2000) As Double
Dim order(1 To 4, 1 To 2000) As Double
Dim onorder(1 To 4) As Integer
Dim shipment(1 To 4, 1 To 2000) As Double
Dim basestock(1 To 4) As Double
Dim leadtime(1 To 4) As Double
Dim invshipped(1 To 4) As Double
Dim z(1 To 4) As Double
Dim disr(1 To 4) As Integer
Dim dtime(1 To 4) As Integer
Dim y(1 To 4) As Double
Dim Inp(1 To 4, 1 To 4) As Integer
Dim Outp(1 To 4, 1 To 10) As Integer
Dim OutpTime(1 To 4, 0 To 10) As Double
Dim OutpTimeLong(1 To 4, 1 To 10) As Double
Dim p(1 To 4, 1 To 4, 1 To 4) As Integer    'Path(number, node, path)
Dim Pcount(1 To 4) As Integer
Dim DT(1 To 4, 1 To 4) As Integer          ' disruption time
Dim Starved(1 To 4, 1 To 4) As Integer
Dim initial(1 To 4, 1 To 4) As Integer
Dim timecheck(1 To 8, 1 To 8) As Integer
Dim DR(1 To 4, 1 To 4) As Integer
Dim dc(1 To 4, 1 To 4) As Integer
Dim count(1 To 4) As Integer
Dim effected(1 To 4) As Integer
Dim POS(1 To 4) As Integer
Dim check2(1 To 4) As Integer
Dim check(1 To 4) As Integer
Dim DS(1 To 4, 1 To 4) As Integer
```

```
Dim down(1 To 4, 1 To 4) As Integer
Dim i, j, k, startnode, node, row1, column1, check1, l As Integer
Dim row, column, m, n, bf, review, pois As Integer
Dim x, time, delta, lt, dummy, canship1, canship2, canship3, deamandloss, mu3, std3, csl, csl1, short0, short2, short3, howsoon,
howlong As Double
leadtime(2) = Worksheets("SC Williem and Graves v1").Cells(15, 3)
leadtime(3) = Worksheets("SC Williem and Graves v1").Cells(13, 3)
leadtime(4) = Worksheets("SC Williem and Graves v1").Cells(11, 3)
mu3 = Worksheets("SC Williem and Graves v1").Cells(18, 10)
std3 = Worksheets("SC Williem and Graves v1").Cells(18, 11)
inventory(4) = Worksheets("SC Williem and Graves v1").Cells(28, 8)  'in units
inventory(3) = Worksheets("SC Williem and Graves v1").Cells(29, 8)
inventory(2) = Worksheets("SC Williem and Graves v1").Cells(30, 8)
basestock(4) = Worksheets("SC Williem and Graves v1").Cells(28, 8)  'in units
basestock(3) = Worksheets("SC Williem and Graves v1").Cells(29, 8)
basestock(2) = Worksheets("SC Williem and Graves v1").Cells(30, 8)
For k = 1 To 1010
   For m = 1 To 4
      order(m, k) = 0
      shipment(m, k) = 0
   Next
Next
For k = 0 To 1010
   For m = 0 To 4
      unsat(m, k) = 0
   Next
Next
For i = 1 To 4
   dtime(i) = 0
   effected(i) = 0
Next
demandloss = 0
review = 1
row = 33
csl = 0
csl1 = 0
For i = 1 To 4
```

```
    For m = 1 To 4
        DT(i, m) = 0
    Next
Next
'Reachability Analysis
row1 = 57
column1 = 28
For i = 1 To 4                'Fill the Inp Matrix
    For j = 1 To 4
        Inp(i, j) = Worksheets("22 Node SC").Cells(row1, column1)
        column1 = column1 + 1
    Next
    row1 = row1 + 1
    column1 = 28
Next
column1 = 1
k = 1
Pcount(1) = 1
p(1, 1, 1) = 1
For i = 1 To 4                'Fill in the paths
    For j = 1 To 4
        If Inp(i, j) = 1 Then
            Pcount(i) = Pcount(i) + 1
            m = 1
            p(Pcount(i), i, m) = i
            For n = 1 To Pcount(j)
                While p(n, j, m) <> 0
                    p(Pcount(i), i, m + 1) = p(n, j, m)
                    m = m + 1
                Wend
                m = 1
                If n < Pcount(j) Then Pcount(i) = Pcount(i) + 1
            Next
            If p(Pcount(i), i, 1) = 0 Then p(Pcount(i), i, 1) = i
        End If
    Next
Next
```

```
'MsgBox "Path for node 4 is  " & P(1, 4, 1) & " " & P(1, 4, 2) & " " & P(1, 4, 3)
For i = 1 To 4                          'Perform Reachability Analysis/Fill the Outp Matrix
   For j = 1 To 4
     If Inp(i, j) = 1 Then
       Outp(i, column1) = j
       k = 1
       While Outp(j, k) <> 0
          Outp(i, column1 + 1) = Outp(j, k)
          column1 = column1 + 1
          k = k + 1
       Wend
     column1 = column1 + 1
     End If
   Next
column1 = 1
Next
column1 = 1
row1 = 66
column1 = 28
For i = 1 To 4                        'Out put the Outp Matrix into the Excel
   For j = 1 To 4
      Worksheets("22 Node SC").Cells(row1, column1).Value = Outp(i, j)
      column1 = column1 + 1
   Next
row1 = row1 + 1
column1 = 28
Next
For row1 = 66 To 69                   'Remove Zeros from Outp Matrix
   For column1 = 28 To 31
      If Worksheets("22 Node SC").Cells(row1, column1) = 0 Then
         If row1 <> 67 Or column1 <> 28 Then
            Worksheets("22 Node SC").Cells(row1, column1) = Empty
               'Selection.ClearContents
         End If
      End If
   Next
Next
```

```
'End Reachability Analysis
'---------------------------------------Start time simulation-----------------------------
For time = 1 To 1000
'-----------------------------That's where RA used to be----------------------------------
'--------------------------------------------------------------------------------------------
Sheets("Sheet1").Select
   column = 1
   Worksheets("Sheet1").Cells(row, column).Value = time
'Fill in lead time
   leadtime(2) = Worksheets("SC Williem and Graves v1").Cells(15, 3)
   leadtime(3) = Worksheets("SC Williem and Graves v1").Cells(13, 3)
   leadtime(4) = Worksheets("SC Williem and Graves v1").Cells(11, 3)
   For i = 2 To 4
      If disr(i) = 1 Then
         leadtime(i) = leadtime(i) + y(i)
         y(i) = Application.WorksheetFunction.Max(0, y(i) - 1)
      End If
   Next
'Node 1 operational Logic ----------------------------------------------------------------------------------------------
   inventory(2) = inventory(2) + shipment(2, time)         'begining inventory
   onorder(2) = onorder(2) - shipment(2, time)
'update unsat vector
   n = 15
   For k = 1 To 15
      For m = 3 To 15            'Searching for appropriate delta
         If Worksheets("Demand Decay").Cells(m, 4).Value = n - k + 1 Then delta = Worksheets("Demand Decay").Cells(m, 5).Value
      Next
      unsat(0, n - k + 1) = delta * unsat(0, n - k)
      unsat(1, n - k + 1) = delta * unsat(1, n - k)
      demandloss = demandloss + unsat(0, n - k) * (1 - delta)
      unsat(0, n - k) = 0
      unsat(1, n - k) = 0
   Next
   Worksheets("Sheet1").Cells(row, column + 14).Value = demandloss
   x = Rnd()
   D(2) = Application.WorksheetFunction.Max(0, Application.WorksheetFunction.NormInv(x, mu3, std3))
   Worksheets("Sheet1").Cells(row, column + 1).Value = D(2)
```

```
'checking how much of backlog can be satisfied
    n = 15
    For k = 0 To 15
        canship2 = inventory(2)
        inventory(2) = inventory(2) - Application.WorksheetFunction.Min(unsat(0, n - k), canship2)
        unsat(0, n - k) = unsat(0, n - k) - Application.WorksheetFunction.Min(unsat(0, n - k), canship2)
        short0 = short0 + unsat(0, n - k)
    Next
'Determine How much of the current inventory can be shipped
    canship2 = inventory(2)
    inventory(2) = inventory(2) - Application.WorksheetFunction.Min(D(2), canship2)
    unsat(0, 0) = D(2) - Application.WorksheetFunction.Min(D(2), canship2)
    short0 = short0 + unsat(0, 0)
    If unsat(0, 0) > 0 Then csl = csl + 1
    Worksheets("Sheet1").Cells(row, column + 2).Value = inventory(2)
    Worksheets("Sheet1").Cells(row, column + 4).Value = unsat(0, 0)
    order(2, time) = D(2) 'basestock(1) - inventory(1) - onorder(1) + unsat(0, 0)
    Worksheets("Sheet1").Cells(row, column + 5).Value = order(2, time)
    onorder(2) = onorder(2) + order(2, time)

'Node 1 operational logic-------------------------------------------------------------------------------------

    inventory(4) = inventory(4) + shipment(4, time)   'begining inventory at node 1 for supplier 2
    inventory(3) = inventory(3) + shipment(3, time)   'begining inventory at node 1 for supplier 3
    onorder(4) = onorder(4) - shipment(4, time)
    onorder(3) = onorder(3) - shipment(3, time)

'checking how much of backlog can be satisfied
    For k = 0 To 15
        canship2 = Application.WorksheetFunction.Min(inventory(4), inventory(3))
        shipment(2, time + leadtime(2)) = shipment(2, time + leadtime(2)) + Application.WorksheetFunction.Min(unsat(1, n - k),
canship2)
        inventory(4) = inventory(4) - (Application.WorksheetFunction.Min(unsat(1, n - k), canship2))
        inventory(3) = inventory(3) - (Application.WorksheetFunction.Min(unsat(1, n - k), canship2))
        unsat(1, n - k) = unsat(1, n - k) - Application.WorksheetFunction.Min(unsat(1, n - k), canship2)
        short4 = short4 + unsat(1, n - k)
        short3 = short3 + unsat(1, n - k)
```

```
    Next
'Determine how much can be shiped of the current order
    canship2 = Application.WorksheetFunction.Min(inventory(4), inventory(3))
    shipment(2, time + leadtime(2)) = shipment(2, time + leadtime(2)) + Application.WorksheetFunction.Min(D(2), canship2)
    Worksheets("Sheet1").Cells(row + leadtime(2), column + 3).Value = shipment(2, time + leadtime(2))
    unsat(1, 0) = D(2) - Application.WorksheetFunction.Min(D(2), canship2)
    If unsat(1, 0) > 0 Then csl1 = csl1 + 1
    short4 = short4 + unsat(1, 0)
    short3 = short3 + unsat(1, 0)
    Worksheets("Sheet1").Cells(row, column + 8) = unsat(1, 0)
    Worksheets("Sheet1").Cells(row, column + 12) = unsat(1, 0)
'Ending Inventory for buffers 2 and 3
    inventory(4) = inventory(4) - (Application.WorksheetFunction.Min(D(2), canship2))   'Ending inventory
    inventory(3) = inventory(3) - (Application.WorksheetFunction.Min(D(2), canship2))      'Ending inventory
    Worksheets("Sheet1").Cells(row, column + 6).Value = inventory(3)
    Worksheets("Sheet1").Cells(row, column + 10).Value = inventory(4)
'Order 2 shippment
    order(3, time) = D(2) 'basestock(3) - inventory(3) - onorder(3) + unsat(1, 0) * p
    shipment(3, time + leadtime(3)) = shipment(3, time + leadtime(3)) + order(3, time)
    Worksheets("Sheet1").Cells(row + leadtime(3), column + 7).Value = shipment(3, time + leadtime(3))
    onorder(3) = onorder(3) + order(3, time)
    Worksheets("Sheet1").Cells(row, column + 9).Value = order(3, time)
'Order 3 shipment
    order(4, time) = D(2) 'basestock(4) - inventory(4) - onorder(4) + unsat(1, 0)
    shipment(4, time + leadtime(4)) = shipment(4, time + leadtime(4)) + order(4, time)
    Worksheets("Sheet1").Cells(row + leadtime(4), column + 11).Value = shipment(4, time + leadtime(4))
    onorder(4) = onorder(4) + order(4, time)
    Worksheets("Sheet1").Cells(row, column + 13).Value = order(4, time)

'---------------------------------Beginning of RA and disruption----------------------------

'Actual TRA
    For i = 2 To 4
            For k = 1 To Pcount(i)
                For m = 1 To 4

                    If p(k, i, m) > 1 Then
```

```
    If DS(i, p(k, i, m)) = 1 Or DR(i, p(k, i, m)) = 1 Then

        If inventory(p(k, i, m)) = 0 Then
            DS(i, p(k, i, m + 1)) = 1
            If initial(i, p(k, i, m + 1)) = 0 And DS(i, p(k, i, m + 1)) Then
                Starved(i, p(k, i, m + 1)) = Starved(i, p(k, i, m + 1)) + 1
                initial(i, p(k, i, m + 1)) = 1
                'MsgBox "in"
            End If
        End If
         If DS(i, p(k, i, m)) = 1 Then
            If m = 1 Then DT(i, p(k, i, m)) = DT(i, p(k, i, m)) + 1
            If m > 1 Then
                If inventory(p(k, i, m - 1)) = 0 Then DT(i, p(k, i, m)) = DT(i, p(k, i, m)) + 1
            End If
         End If
         'DT(i, P(k, i, m)) = DT(i, P(k, i, m)) + 1
            'timecheck(i,h P(k, i, m)) = 1
        'End If
    End If
End If
If p(k, i, m) = 1 Then
    If DS(i, p(k, i, m)) = 1 And inventory(p(k, i, m - 1)) = 0 Then DT(i, p(k, i, m)) = DT(i, p(k, i, m)) + 1
End If
If m = 1 Then
    If time = dtime(i) Then
        DS(i, i) = 0
        DR(i, i) = 1
        dc(i, i) = time + leadtime(i)
    End If
    If time = dc(i, i) Then
        DR(i, i) = 0
        dc(i, i) = 0
    End If
End If
If m > 1 And p(k, i, m) > 0 Then
    If DS(i, p(k, i, m)) = 1 And inventory(p(k, i, m - 1)) > 0 Then
```

```
                    DS(i, p(k, i, m)) = 0
                    DR(i, p(k, i, m)) = 1
                    dc(i, p(k, i, m)) = time + leadtime(p(k, i, m))
                End If
                If time = dc(i, p(k, i, m)) Then
                    DR(i, p(k, i, m)) = 0
                    initial(i, p(k, i, m)) = 0
                    dc(i, p(k, i, m)) = 0
                End If
            End If
        Next
    Next
Next
'End of TRA
Next
'End of TRA
For i = 1 To 4
    check(i) = 0
Next
'Disruption Simulation
For i = 2 To 4
    If disr(i) = 0 Then z(i) = Rnd()
Next
If z(2) < Worksheets("SC Williem and Graves v1").Cells(14, 4) Then
    y(2) = Rnd() * (Worksheets("SC Williem and Graves v1").Cells(15, 4) - Worksheets("SC Williem and Graves v1").Cells(15, 3) -
1) + 1
    y(2) = Application.WorksheetFunction.Round(y(2), 0)
    disr(2) = 1
    dtime(2) = time + y(2)
    z(2) = 0.6
    For i = time + 1 To i = time + leadtime(2)
        invshipped(2) = invshipped(2) + shipment(2, i)
    Next
    check2(2) = 1
    DS(2, 2) = 1
    Starved(2, 2) = Starved(2, 2) + 1
    count(2) = count(2) + 1
End If
If z(3) < Worksheets("SC Williem and Graves v1").Cells(12, 4) Then
```

```
        y(3) = Rnd() * (Worksheets("SC Williem and Graves v1").Cells(13, 4) - Worksheets("SC Williem and Graves v1").Cells(13, 3) -
1) + 1
        y(3) = Application.WorksheetFunction.Round(y(3), 0)
        disr(3) = 1
        dtime(3) = time + y(3)
        z(3) = 0.6
        For i = time + 1 To i = time + leadtime(3)
            invshipped(3) = invshipped(3) + shipment(3, i)
        Next
        count(3) = count(3) + 1
        DS(3, 3) = 1
        check2(3) = 1
        'DT(3, 3) = DT(3, 3) + y(3)
        Starved(3, 3) = Starved(3, 3) + 1
    End If
    If z(4) < Worksheets("SC Williem and Graves v1").Cells(10, 4) Then
        y(4) = Rnd() * (Worksheets("SC Williem and Graves v1").Cells(11, 4) - Worksheets("SC Williem and Graves v1").Cells(11, 3) -
1) + 1
        y(4) = Application.WorksheetFunction.Round(y(4), 0)
        disr(4) = 1
        dtime(4) = time + y(4)
        z(4) = 0.6
        For i = time + 1 To i = time + leadtime(4)
            invshipped(4) = invshipped(4) + shipment(4, i)
        Next
        count(4) = count(4) + 1
        check2(4) = 1
        DS(4, 4) = 1
        'DT(4, 4) = DT(4, 4) + y(4)
        Starved(4, 4) = Starved(4, 4) + 1
    End If
'Time Reachability Analysis
    For i = 2 To 4
        If check2(i) = 1 Then
            check2(i) = 0
            howsoon = 0
            howlong = 0
```

```
            row1 = 73
            column1 = 28
            For l = 1 To 4
                POS(l) = inventory(l) / 200 + onorder(l) / 200 '(lApplication.WorksheetFunction.Max(invshipped(l) / 200, leadtime(l) - 1)
'invshipped(l)
            Next
            k = 1
            howlong = y(i)
            howsoon = howsoon + POS(i)
            OutpTimeLong(i, i) = howlong
            howlong = howlong + leadtime(i) - POS(i)
                'effected(i) = effected(i) + 1
            While Outp(i, k) <> 0
                j = Outp(i, k)
                If howlong <= 0 Then
                    howlong = 0
                    howsoon = 0
                End If
                If howlong > 0 Then
                    effected(j) = effected(j) + 1
                End If
                OutpTime(i, j) = howsoon
                OutpTimeLong(i, j) = howlong
                howlong = howlong + leadtime(j) - POS(j)
                howsoon = howsoon + POS(j)
                k = k + 1
            Wend
            For j = 1 To 4
                Worksheets("22 Node SC").Cells(row1 + i, column1).Value = Worksheets("22 Node SC").Cells(row1 + i, column1).Value
+ OutpTime(i, j)
                Worksheets("22 Node SC").Cells(row1 + i, column1 + 7).Value = Worksheets("22 Node SC").Cells(row1 + i, column1 +
7).Value + OutpTimeLong(i, j)
                column1 = column1 + 1
            Next
        End If
    Next
'End of Time Reachability analysis
```

```
    For i = 1 To 4
        If time = dtime(i) Then
            disr(i) = 0
            dtime(i) = 0
            'invshipped(i) = 0
        End If
    Next

    For l = 1 To 4
        For n = 1 To 4
            timecheck(l, n) = 0
        Next
    Next
'----------------------------------End of RA----------------------------------------------
'Row update
    row = row + 1
Next
Worksheets("Sheet1").Cells(1043, 2).Value = 1 - csl / 1000
Worksheets("Sheet1").Cells(1043, 6).Value = 1 - csl1 / 1000
Worksheets("Sheet1").Cells(1043, 10).Value = 1 - csl1 / 1000
'Worksheets("Sheet1").Cells(1044, 5).Value = short0
'Worksheets("Sheet1").Cells(1044, 9).Value = short3
'Worksheets("Sheet1").Cells(1044, 13).Value = short4
row1 = 78
column1 = 35
For i = 1 To 4
    Worksheets("22 Node SC").Cells(row1, column1) = count(i)
    Worksheets("22 Node SC").Cells(row1 + 1, column1) = effected(i)
    column1 = column1 + 1
Next
i = 1
j = 1
For row = 74 To 77                  'Output Starved matrix
    For column = 42 To 45
        Worksheets("22 Node SC").Cells(row, column).Value = Starved(i, j)
        j = j + 1
    Next
```

```
    j = 1
    i = i + 1
Next
i = 1
j = 1
For row = 74 To 77                  'Output DT matrix
    For column = 48 To 51
        Worksheets("22 Node SC").Cells(row, column).Value = DT(i, j)
        j = j + 1
    Next
    j = 1
    i = i + 1
Next
End Sub
```

Appendix IV

VBA for Four-Node Supply Chain Sensitivity and Scenario Analysis

```
Sub RunTheCode()
Dim run, row, row1, column, h, i, k, j, pr As Integer
Dim p As Double
Dim prob, dev, lt As Double
Dim AveInv(1 To 4) As Double
Dim AIS(1 To 4, 1 To 15) As Double  'AveInv Standard Deviation
Dim SL(1 To 4) As Double
Dim SLS(1 To 4, 1 To 15) As Double  'SL Standard deviantion
Dim FR(1 To 4) As Double
Dim FRS(1 To 4, 1 To 15) As Double  'FR Standard Deviation
Dim Length(1 To 4) As Double
Dim AveDisrCount(1 To 4) As Integer
Dim ADCS(1 To 4, 1 To 15) As Double     'Ave Disr Count Standard Deviation
Dim AveEffected(1 To 4) As Integer
Dim bs(1 To 4) As Integer
Dim sldev(1 To 4) As Double
Dim frdev(1 To 4) As Double
Dim aveinvdev(1 To 4) As Double
Dim adcdev(1 To 4) As Double
prob = 0.98
dev = 30
lt = 9
```

```
k = 13
p = 0.99
row1 = 253            'change here
'While p >= 0.85           'chage here
j = 1
For i = 1 To 4
    AveInv(i) = 0
    SL(i) = 0
    FR(i) = 0
    Length(i) = 0
    AveDisrCount(i) = 0
    AveEffected(i) = 0
    bs(i) = 0
    sldev(i) = 0
    frdev(i) = 0
    aveinvdev(i) = 0
    adcdev(i) = 0
    For k = 1 To 15
        AIS(i, k) = 0
        SLS(i, k) = 0
        FRS(i, k) = 0
        ADCS(i, k) = 0
    Next
Next
'Worksheets("SC Williem and Graves v1").Cells(14, 3) = p        'change here
```

```
Worksheets("SC Williem and Graves v1").Select
SolverOk SetCell:="$M$21", MaxMinVal:=2, ValueOf:="0", ByChange:="$C$23:$C$25"
SolverSolve UserFinish:=True
'MsgBox Worksheets("SC Williem and Graves v1").Cells(19, 7)
For run = 1 To 10
    Application.run "'VBA Thesis.xls'!ThisWorkbook.clearthetable"
    Application.run "'VBA Thesis.xls'!ThreeNodeSCTRA.ThreeNodeSC"
'----------------------Average End. Inv, Sl, FR----------------------------------------
    column = 3
    For i = 1 To 3
        AveInv(i) = AveInv(i) + Worksheets("Sheet1").Cells(1046, column)
        AIS(i, j) = Worksheets("Sheet1").Cells(1046, column)
        SL(i) = SL(i) + Worksheets("Sheet1").Cells(1043, column - 1)
        SLS(i, j) = Worksheets("Sheet1").Cells(1043, column - 1)
        FR(i) = FR(i) + Worksheets("Sheet1").Cells(1044, column - 1)
        FRS(i, j) = Worksheets("Sheet1").Cells(1044, column - 1)
        column = column + 4
    Next
'--------------------------------Length---------------------------------------------------
    row = 75
    For i = 1 To 3
        Length(i) = Length(i) + Worksheets("22 Node SC").Cells(row, 48)
        AveEffected(i) = AveEffected(i) + Worksheets("22 Node SC").Cells(row, 42)
        row = row + 1
    Next
```

```
'----------------------------------Base Stock---------------------------------------------
    row = 30
    For i = 1 To 3
        bs(i) = bs(i) + Worksheets("SC Williem and Graves v1").Cells(row, 8)
        row = row - 1
    Next
'---------------------AveDisrCount---------------------------------------------------------
    column = 36
    For i = 1 To 3
        AveDisrCount(i) = AveDisrCount(i) + Worksheets("22 Node SC").Cells(78, column).Value
        ADCS(i, j) = Worksheets("22 Node SC").Cells(78, column).Value
        'AveEffected(i) = AveEffected(i) + Worksheets("22 Node SC").Cells(79, column - 1).Value
        column = column + 1
    Next
    j = j + 1
Next

For i = 1 To 3
    For j = 1 To 10
        sldev(i) = sldev(i) + (SLS(i, j) - SL(i) / (run - 1)) ^ 2
        frdev(i) = frdev(i) + (FRS(i, j) - FR(i) / (run - 1)) ^ 2
        aveinvdev(i) = aveinvdev(i) + (AIS(i, j) - AveInv(i) / (run - 1)) ^ 2
        adcdev(i) = adcdev(i) + (ADCS(i, j) - AveDisrCount(i) / (run - 1)) ^ 2
    Next
Next
```

```
column = 1
For i = 1 To 3
    sldev(i) = 1.833 * (sldev(i) / (run - 2)) ^ (1 / 2)
    frdev(i) = 1.833 * (frdev(i) / (run - 2)) ^ (1 / 2)
    aveinvdev(i) = 1.833 * (aveinvdev(i) / (run - 2)) ^ (1 / 2)
    adcdev(i) = 1.833 * (adcdev(i) / (run - 2)) ^ (1 / 2)
Next

column = 3
For i = 1 To 3
    Worksheets("Sensitivity").Cells(row1, column) = AveInv(i) / (run - 1)
    Worksheets("Sensitivity").Cells(row1, column + 3) = aveinvdev(i)
    Worksheets("Sensitivity").Cells(row1, column + 6) = SL(i) / (run - 1)
    Worksheets("Sensitivity").Cells(row1, column + 9) = sldev(i)
    Worksheets("Sensitivity").Cells(row1, column + 12) = FR(i) / (run - 1)
    Worksheets("Sensitivity").Cells(row1, column + 15) = frdev(i)
    Worksheets("Sensitivity").Cells(row1, column + 18) = AveDisrCount(i) / (run - 1)
    Worksheets("Sensitivity").Cells(row1, column + 21) = adcdev(i)
    Worksheets("Sensitivity").Cells(row1, column + 24) = AveEffected(i) / (run - 1)
    Worksheets("Sensitivity").Cells(row1, column + 27) = Length(i) / (run - 1)
    Worksheets("Sensitivity").Cells(row1, column + 30) = bs(i) / (run - 1)
    column = column + 1
Next
```

```
'Worksheets("Sensitivity").Cells(row1, 2) = p       'change here
'p = p - 0.01                                        'change here
row1 = row1 + 1
'Wend
End Sub
```

Appendix V

VBA for Eight-Node Supply Chain Simulation

```
Sub EightNodeSC()
Dim inventory(1 To 9) As Double
Dim basestock(1 To 9) As Double
Dim onorder(1 To 8) As Double
Dim order(1 To 8, 1 To 2000) As Double
Dim shipment(1 To 8, 1 To 2000) As Double
Dim unsat(0 To 9, 0 To 2000) As Double
Dim short(1 To 9) As Double
Dim SL(1 To 8) As Double
Dim leadtime(1 To 8) As Double
Dim z(1 To 8) As Double
Dim disr(1 To 8) As Integer
Dim dtime(1 To 8) As Integer
Dim y(1 To 8) As Double
Dim Inp(1 To 9, 1 To 8) As Integer
Dim Outp(1 To 8, 1 To 100) As Integer
Dim OutpTime(1 To 8, 0 To 8) As Double
Dim OutpTimeLong(1 To 8, 1 To 8) As Double
Dim probab(1 To 8) As Double
Dim longleadtime(1 To 8) As Double
Dim count(1 To 8) As Integer
Dim effected(1 To 8) As Integer
```

```
Dim POS(1 To 8) As Integer
Dim check2(1 To 8) As Integer
Dim abovezero(1 To 9) As Double
Dim P(1 To 8, 1 To 8, 1 To 8) As Integer    'Path(number, node, path)
Dim Pcount(1 To 8) As Integer
Dim DT(1 To 8, 1 To 8) As Integer           ' disruption time
Dim Starved(1 To 8, 1 To 8) As Integer
Dim initial(1 To 8, 1 To 8) As Integer
Dim timecheck(1 To 8, 1 To 8) As Integer
Dim DR(1 To 8, 1 To 8) As Integer
Dim dc(1 To 8, 1 To 8) As Integer
Dim DS(1 To 8, 1 To 8) As Integer
Dim down(1 To 8, 1 To 8) As Integer
Dim time, mu, delta, std, x, Demand, canship, dummy As Double
Dim row, column, i, j, l, row2, column2, column1 As Integer
'Fill in the Inp Matrix
row = 50
For i = 1 To 9                  'rows
   column = 2
   For j = 1 To 8               'columns
      Inp(i, j) = Worksheets("Input").Cells(row, column)
      column = column + 1
   Next
   row = row + 1
Next
```

```
'Fill The paths
column1 = 1
k = 1
Pcount(1) = 1
P(1, 1, 1) = 1
For i = 1 To 8                        'Fill in the paths
    For j = 1 To 8
        If Inp(i, j) = 1 Then
            Pcount(i) = Pcount(i) + 1
            m = 1
            P(Pcount(i), i, m) = i
            For n = 1 To Pcount(j)
                While P(n, j, m) <> 0
                    P(Pcount(i), i, m + 1) = P(n, j, m)
                    m = m + 1
                Wend
                m = 1
                If n < Pcount(j) Then Pcount(i) = Pcount(i) + 1
            Next
            If P(Pcount(i), i, 1) = 0 Then P(Pcount(i), i, 1) = i
        End If
    Next
Next
'Fill inventory and basestock information
row = 44
```

```
column = 8
For i = 2 To 8
    inventory(i) = Worksheets("Input").Cells(row, column)
    basestock(i) = Worksheets("Input").Cells(row, column)
    row = row - 1
Next
'Fill in disruption probability information
row = 15
column = 4
For i = 2 To 8
    probab(i) = Worksheets("Input").Cells(row, column)
    row = row - 2
Next
'Fill in leadtime information
row = 16
column = 3
For i = 2 To 8
    leadtime(i) = Worksheets("Input").Cells(row, column)
    row = row - 2
Next
'Fill in long lead time info
row = 16
column = 4
For i = 2 To 8
    longleadtime(i) = Worksheets("Input").Cells(row, column)
```

```
    row = row - 2
Next
'Initialize various parameters
inventory(9) = 1E+17
basestock(9) = 1E+17
leadtime(1) = 0
mu = Worksheets("Input").Cells(26, 10)
std = Worksheets("Input").Cells(26, 11)

'Begining of Reachability Analysis-------------------------------------------------------
column1 = 1
k = 1
For i = 1 To 8                          'Perform Reachability Analysis/Fill the Outp Matrix
    For j = 1 To 8
      If Inp(i, j) = 1 Then
        Outp(i, column1) = j
        k = 1
        While Outp(j, k) <> 0
            Outp(i, column1 + 1) = Outp(j, k)
            column1 = column1 + 1
            k = k + 1
        Wend
      column1 = column1 + 1
      End If
    Next
```

```
column1 = 1
Next
column1 = 1
row1 = 1032
column1 = 2
For i = 1 To 8                          'Out put the Outp Matrix into the Excel
   For j = 1 To 8
      Worksheets("Output").Cells(row1, column1).Value = Outp(i, j)
      column1 = column1 + 1
   Next
row1 = row1 + 1
column1 = 2
Next
For row1 = 1032 To 1039                 'Remove Zeros from Outp Matrix
   For column1 = 2 To 9
      If Worksheets("Output").Cells(row1, column1) = 0 Then
         If row1 <> 1032 Or column1 <> 2 Then
            Worksheets("Output").Cells(row1, column1) = Empty
         End If
      End If
   Next
Next
'End of Reachability Analysis---------------------------------------------------------------

'Start the simulation---------------------------------------------------------------------
```

```
row2 = 16
For time = 1 To 1000
    column2 = 3
    Worksheets("Output").Cells(row2, column2 - 2).Value = time
'Fill in Leadtime----------------------------------------------------------------------------
    row = 16
    column = 3
    For l = 2 To 8
        leadtime(l) = Worksheets("Input").Cells(row, column)
        row = row - 2
    Next

    For node = 2 To 8
        If disr(node) = 1 Then
            leadtime(node) = leadtime(node) + y(node)
            y(node) = Application.WorksheetFunction.Max(0, y(node) - 1)
        End If
    Next
'Operational Logic------------------------------------------------------------------------
    x = Rnd()
    Demand = Application.WorksheetFunction.NormInv(x, mu, std)
    Worksheets("Output").Cells(row2, column2 - 1).Value = Demand
    For i = 1 To 8                                  'Account for Beg. Inventory and OO
        For l = 1 To 8
            If Inp(l, i) = 1 Then
```

```
            inventory(l) = inventory(l) + shipment(l, time)
            onorder(l) = onorder(l) - shipment(l, time)
        End If
    Next
'---------------update unsat vector-------------------------------------------------
    n = 15
    For k = 1 To 15
        For m = 3 To 15             'Searching for appropriate delta
            If Worksheets("Demand Decay").Cells(m, 4).Value = n - k + 1 Then delta = Worksheets("Demand Decay").Cells(m, 5).Value
        Next
        unsat(i, n - k + 1) = delta * unsat(i, n - k)
        'demandloss = demandloss + unsat(0, n - k) * (1 - delta)
        unsat(i, n - k) = 0
    Next
'------------------Check how much of backlog can be satisfied------------------------------
    n = 15
    For k = 0 To 15
        For l = 1 To 9
            If Inp(l, i) = 0 Then abovezero(l) = 9000000
            If Inp(l, i) = 1 Then abovezero(l) = inventory(l)
        Next
        canship = Application.WorksheetFunction.Min(abovezero(1), abovezero(2), abovezero(3), abovezero(4), abovezero(5), abovezero(6), abovezero(7), abovezero(8))
```

```
            shipment(i, time + leadtime(i)) = shipment(i, time + leadtime(i)) + Application.WorksheetFunction.Min(unsat(i, n - k), canship)
            For l = 1 To 9
              If Inp(l, i) = 1 Then
                inventory(l) = inventory(l) - Application.WorksheetFunction.Min(unsat(i, n - k), canship)
                'short(l) = short(l) + unsat(l, n - k)
              End If
            Next
            unsat(i, n - k) = unsat(i, n - k) - Application.WorksheetFunction.Min(unsat(i, n - k), canship)
          Next
          For l = 1 To 8
            abovezero(l) = 0
          Next
    '--------------------Satisfy current period Demand------------------------------
          For l = 1 To 9
            If Inp(l, i) = 0 Then abovezero(l) = 900000000
            If Inp(l, i) = 1 Then abovezero(l) = inventory(l)
          Next
          canship = Application.WorksheetFunction.Min(abovezero(1), abovezero(2), abovezero(3), abovezero(4), abovezero(5), abovezero(6), abovezero(7), abovezero(8))
          shipment(i, time + leadtime(i)) = shipment(i, time + leadtime(i)) + Application.WorksheetFunction.Min(Demand, canship)
          Worksheets("Output").Cells(row2 + leadtime(i), column2 + 1).Value = shipment(i, time + leadtime(i))
          column1 = 3
          For l = 1 To 9
            If Inp(l, i) = 1 Then
```

```
            inventory(l) = inventory(l) - Application.WorksheetFunction.Min(Demand, canship)
            short(l) = short(l) + unsat(l, 0)
            Worksheets("Output").Cells(row2, column1).Value = inventory(l)
        End If
        column1 = column1 + 4
    Next
    'Worksheets("Output").Cells(row2, column2).Value = inventory(i)
    unsat(i, 0) = Demand - Application.WorksheetFunction.Min(Demand, canship)
    Worksheets("Output").Cells(row2, column2 + 2).Value = unsat(i, 0)
    If unsat(i, 0) > 0 Then SL(i) = SL(i) + 1
    order(i, time) = Demand 'basestock(1) - inventory(1) - onorder(1) + unsat(0, 0)
    Worksheets("Output").Cells(row2, column2 + 3).Value = order(i, time)
    onorder(i) = onorder(i) + order(i, time)
    For l = 1 To 8
        abovezero(l) = 0
    Next
    column2 = column2 + 4
Next 'end of cycling through the nodes

'-----------------------------------Disruption Simulation And RA-----------------------------
'--------------------------Actual RA------------------------------------------------------
For i = 2 To 8
        For k = 1 To Pcount(i)
            For m = 1 To 8
```

```
If P(k, i, m) > 1 Then
    If DS(i, P(k, i, m)) = 1 Or DR(i, P(k, i, m)) = 1 Then
        If inventory(P(k, i, m)) = 0 Then
        'MsgBox "node is " & P(k, i, m) & " inv.  " & inventory(P(k, i, m)) & " time is " & time & " DS(2, 1)= " & DS(i,
P(k, i, m + 1))
            DS(i, P(k, i, m + 1)) = 1
            If initial(i, P(k, i, m + 1)) = 0 And DS(i, P(k, i, m + 1)) Then
                Starved(i, P(k, i, m + 1)) = Starved(i, P(k, i, m + 1)) + 1
                initial(i, P(k, i, m + 1)) = 1
            End If
        End If
        'If timecheck(i, P(k, i, m)) = 0 And DS(i, P(k, i, m)) = 1 Then
         If DS(i, P(k, i, m)) = 1 Then
            If m = 1 Then DT(i, P(k, i, m)) = DT(i, P(k, i, m)) + 1
            If m > 1 Then
                If inventory(P(k, i, m - 1)) = 0 Then DT(i, P(k, i, m)) = DT(i, P(k, i, m)) + 1
            End If
        End If
            'timecheck(i,h P(k, i, m)) = 1
        'End If
    End If
End If
If P(k, i, m) = 1 Then
    If DS(i, P(k, i, m)) = 1 And inventory(P(k, i, m - 1)) = 0 Then
        DT(i, P(k, i, m)) = DT(i, P(k, i, m)) + 1
```

```
    End If
End If

If m = 1 Then
    If time = dtime(i) Then
        DS(i, i) = 0
        DR(i, i) = 1
        dc(i, i) = time + leadtime(i)
    End If
    If time = dc(i, i) Then
        DR(i, i) = 0
        dc(i, i) = 0
    End If
End If
If m > 1 And P(k, i, m) > 0 Then
    If DS(i, P(k, i, m)) = 1 And inventory(P(k, i, m - 1)) > 0 Then
        DS(i, P(k, i, m)) = 0
        DR(i, P(k, i, m)) = 1
        dc(i, P(k, i, m)) = time + leadtime(P(k, i, m))
    End If
    If time = dc(i, P(k, i, m)) Then
        DR(i, P(k, i, m)) = 0
        initial(i, P(k, i, m)) = 0
        dc(i, P(k, i, m)) = 0
    End If
```

```
            End If
          Next
        Next
    Next
'--------------------------End Of Actual RA---------------------------------------------
'Begin Disruption Simulation-----------------------------------------------------------------
    For node = 2 To 8
        If disr(node) = 0 Then z(node) = Rnd()
    Next
    For node = 2 To 8
        If z(node) < probab(node) Then
            y(node) = Rnd() * (longleadtime(node) - leadtime(node) - 1) + 1
            y(node) = Application.WorksheetFunction.Round(y(node), 0)
            disr(node) = 1
            dtime(node) = time + y(node)
            z(node) = 0.6
            check2(node) = 1
            DS(node, node) = 1
            Starved(node, node) = Starved(node, node) + 1
            count(node) = count(node) + 1

        End If
    Next
'Begin Timed Petri Net Reachability-----------------------------------------------------------
    For node = 2 To 8
```

```
If check2(node) = 1 Then
    check2(node) = 0
    howsoon = 0
    howlong = 0
    row1 = 1042
    column1 = 2
    For l = 1 To 8
        POS(l) = (inventory(l) + onorder(l)) / 200
    Next
    k = 1
    howlong = y(node)
    howsoon = howsoon + POS(node)
    OutpTimeLong(node, node) = howlong
    howlong = howlong + leadtime(node) - POS(node)
        'effected(i) = effected(i) + 1
    While Outp(node, k) <> 0
        j = Outp(node, k)
        If howlong <= 0 Then
            howlong = 0
            howsoon = 0
        End If
        If howlong > 0 Then
            effected(j) = effected(j) + 1
        End If
        OutpTime(node, j) = howsoon
```

```
            OutpTimeLong(node, j) = howlong
            howlong = howlong + leadtime(j) - POS(j)
            howsoon = howsoon + POS(j)
            k = k + 1
        Wend
        For j = 1 To 8
            Worksheets("Output").Cells(row1 + node, column1).Value = Worksheets("Output").Cells(row1 + node, column1).Value
+ OutpTime(node, j)
            Worksheets("Output").Cells(row1 + node, column1 + 10).Value = Worksheets("Output").Cells(row1 + node, column1 +
10).Value + OutpTimeLong(node, j)
            column1 = column1 + 1
        Next
      End If
    Next
  'End Timed Petri Net Reachability-----------------------------------------------------------
    For l = 1 To 8
      If time = dtime(l) Then
        disr(l) = 0
        dtime(l) = 0
      End If
    Next

    For l = 1 To 4
      For n = 1 To 4
        timecheck(l, n) = 0
```

```
        Next
    Next

'----------------------------------End of Disruption Simulation And RA----------------------
    row2 = row2 + 1
Next 'end of simulation
row2 = 1023
column2 = 7
For l = 1 To 7
    Worksheets("Output").Cells(row2, column2).Value = 1 - SL(l) / 1000
    column2 = column2 + 4
Next
column2 = 12
For l = 1 To 8
    Worksheets("Output").Cells(1051, column2).Value = count(l)
    Worksheets("Output").Cells(1052, column2).Value = effected(l)
    column2 = column2 + 1
Next
i = 1
j = 1
For row = 1043 To 1050                'Output Starved matrix
    For column = 22 To 29
        Worksheets("Output").Cells(row, column).Value = Starved(i, j)
        j = j + 1
    Next
```

```
   j = 1
   i = i + 1
Next
i = 1
j = 1
For row = 1043 To 1050                'Output DT matrix
   For column = 32 To 39
      Worksheets("Output").Cells(row, column).Value = DT(i, j)
      j = j + 1
   Next
   j = 1
   i = i + 1
Next
End Sub
```

Appendix VI

VBA for Eight-Node Supply Chain Sensitivity and Scenario Analysis

```
Sub RunTheCode()
Dim run, row, row1, column, h, i, j, k As Integer
Dim prob, dev, lt As Double
Dim AveInv(1 To 8) As Double
Dim SL(1 To 8) As Double
Dim SLS(1 To 8, 1 To 11) As Double
Dim sldev(1 To 8) As Double
Dim FR(1 To 8) As Double
Dim FRS(1 To 8, 1 To 11) As Double
Dim frdev(1 To 8) As Double
Dim Length(1 To 8) As Double
Dim AveDisrCount(1 To 8) As Integer
Dim ADCS(1 To 8, 1 To 11) As Double
Dim adcdev(1 To 8) As Double
Dim AveEffected(1 To 8) As Double
Dim bs(1 To 8) As Double
prob = 0.98
dev = 30
lt = 8
k = 13
h = 6
row1 = 6          'change here
```

```
j = 1
'While k >= 3            'chage here
For i = 1 To 7
    AveInv(i) = 0
    SL(i) = 0
    FR(i) = 0
    Length(i) = 0
    AveDisrCount(i) = 0
    AveEffected(i) = 0
    bs(i) = 0
Next
'Worksheets("Input").Cells(11, 4) = k         'change here
Worksheets("Input").Select
SolverOk SetCell:="$M$27", MaxMinVal:=2, ValueOf:="0", ByChange:="$C$29:$C$35"
SolverSolve UserFinish:=True
'MsgBox Worksheets("Input").Cells(19, 7)
For run = 1 To 10
    Application.run "'8 node supply chain.xls'!ClearTheTable.ClearTheTable"
    Application.run "'8 node supply chain.xls'!EightNodeSC.EightNodeSC"
'----------------------Average End. Inv, Sl, FR----------------------------------------
    column = 7
    For i = 1 To 7
        AveInv(i) = AveInv(i) + Worksheets("Output").Cells(1026, column)
        SL(i) = SL(i) + Worksheets("Output").Cells(1023, column).Value
        SLS(i, j) = Worksheets("Output").Cells(1023, column).Value
```

```
        FR(i) = FR(i) + Worksheets("Output").Cells(1024, column).Value
        FRS(i, j) = Worksheets("Output").Cells(1024, column).Value
        column = column + 4
    Next
'---------------------------------Length---------------------------------------------------
    row = 1044
    For i = 1 To 7
        Length(i) = Length(i) + Worksheets("Output").Cells(row, 32)
        AveEffected(i) = AveEffected(i) + Worksheets("Output").Cells(row, 22)
        row = row + 1
    Next
'-----------------------------------Base Stock---------------------------------------------
    row = 44
    For i = 1 To 7
        bs(i) = bs(i) + Worksheets("Input").Cells(row, 8)
        row = row - 1
    Next
'----------------------AveDisrCount, AveEffected-------------------------------------------
    column = 13
    For i = 1 To 7
        AveDisrCount(i) = AveDisrCount(i) + Worksheets("Output").Cells(1051, column).Value
        ADCS(i, j) = Worksheets("Output").Cells(1051, column).Value
        column = column + 1
    Next
    j = j + 1
```

```
Next
For i = 1 To 7
    For j = 1 To 10
        sldev(i) = sldev(i) + (SLS(i, j) - SL(i) / (run - 1)) ^ 2
        frdev(i) = frdev(i) + (FRS(i, j) - FR(i) / (run - 1)) ^ 2
        adcdev(i) = adcdev(i) + (ADCS(i, j) - AveDisrCount(i) / (run - 1)) ^ 2
    Next
Next
column = 1

For i = 1 To 7
    sldev(i) = 1.833 * (sldev(i) / (run - 2)) ^ (1 / 2)
    frdev(i) = 1.833 * (frdev(i) / (run - 2)) ^ (1 / 2)
    adcdev(i) = 1.833 * (adcdev(i) / (run - 2)) ^ (1 / 2)
Next
column = 3
For i = 1 To 7
    Worksheets("Sensitivity").Cells(row1, column) = AveInv(i) / (run - 1)
    Worksheets("Sensitivity").Cells(row1, column + 7) = SL(i) / (run - 1)
    Worksheets("Sensitivity").Cells(row1, column + 14) = sldev(i)
    Worksheets("Sensitivity").Cells(row1, column + 21) = FR(i) / (run - 1)
    Worksheets("Sensitivity").Cells(row1, column + 28) = frdev(i)
    Worksheets("Sensitivity").Cells(row1, column + 35) = AveDisrCount(i) / (run - 1)
    Worksheets("Sensitivity").Cells(row1, column + 42) = adcdev(i)
    Worksheets("Sensitivity").Cells(row1, column + 49) = AveEffected(i) / (run - 1)
```

```
    Worksheets("Sensitivity").Cells(row1, column + 56) = Length(i) / (run - 1)
    Worksheets("Sensitivity").Cells(row1, column + 63) = bs(i) / (run - 1)
    column = column + 1
Next
'Worksheets("Sensitivity").Cells(row1, 2) = k        'change here
k = k - 1                                         'change here
row1 = row1 + 1
'Wend
End Sub
```