

2016

# Strategies for Reducing Supplier Risk: Inputs into the Supply Chain

Christopher A. Greene  
*University of Massachusetts Amherst*

Follow this and additional works at: [https://scholarworks.umass.edu/masters\\_theses\\_2](https://scholarworks.umass.edu/masters_theses_2)

 Part of the [Industrial Engineering Commons](#), [Management and Operations Commons](#), [Other Operations Research, Systems Engineering and Industrial Engineering Commons](#), and the [Risk Analysis Commons](#)

---

## Recommended Citation

Greene, Christopher A., "Strategies for Reducing Supplier Risk: Inputs into the Supply Chain" (2016). *Masters Theses*. 322.  
[https://scholarworks.umass.edu/masters\\_theses\\_2/322](https://scholarworks.umass.edu/masters_theses_2/322)

This Open Access Thesis is brought to you for free and open access by the Dissertations and Theses at ScholarWorks@UMass Amherst. It has been accepted for inclusion in Masters Theses by an authorized administrator of ScholarWorks@UMass Amherst. For more information, please contact [scholarworks@library.umass.edu](mailto:scholarworks@library.umass.edu).

# **STRATEGIES FOR REDUCING SUPPLIER RISK: INPUTS INTO THE SUPPLY CHAIN**

A Thesis Presented  
by  
CHRISTOPHER A. GRENE

Submitted to the Graduate School of the  
University of Massachusetts Amherst in componential fulfillment  
of the requirements for the degree of

**MASTER OF SCIENCE IN INDUSTRIAL ENGINEERING AND OPERATIONS  
RESEARCH**

February 2016

Mechanical and Industrial Engineering

© Copyright by Christopher A. Greene 2016

All Rights Reserved

# STRATEGIES FOR REDUCING SUPPLIER RISK: INPUTS INTO THE SUPPLY CHAIN

A Thesis Presented

by

CHRISTOPHER A. GREENE

Approved as to style and content by:

---

Ana Muriel, Chair

---

Hari Balasubramanian, Member

---

Chaitra Gopalappa, Member

---

Sundar Krishnamurty, Department Head  
Department of Mechanical and Industrial  
Engineering

## ACKNOWLEDEMENTS

Include:

my parents & Jeruska Lugo

Professors Muriel, Balasubramanian and Gopalappa

Michael Prokle and Faried Beladi

Raj Subbu

Professor David Jensen

David Arbour

## ABSTRACT

### STRATEGIES FOR REDUCING SUPPLIER RISK: INPUTS INTO THE SUPPLY CHAIN

FEBRUARY 2016

CHRISTOPHER A. GREENE

M.S.I.E.O.R., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Ana Muriel

There are many aspects to consider when managing an entire supply chain from procurement to fulfillment. Complex assemblies require hundreds of components, sourced from all corners of the globe, to come together in a synchronized fashion. Given the magnitude of the supply chain, high quality standards, and significantly increased outsourcing, there is a strong need to monitor supplier risk and quickly identify and mitigate potential problems. Moreover, the continuous pressure to reduce resources and pressure to cut costs, further increases the need for the development of procedures and tools that can quickly and efficiently address these potential supply chain risks. This thesis focuses on two unique problems brought to our attention by supply chain managers in the field. The first is the analysis of the robustness of advanced ordering strategies (AOS). AOS have been proposed in previous research to coordinate the delivery of components for complex assemblies with long and highly variable lead times. They have been shown to be highly successful to synchronize the supply chain under on-going conditions. It is not clear; however, their effect as the underlying performance of suppliers evolves over time. The second topic covers the methodological foundation and development of a tool to accurately classify suppliers based off risk, and provides a method to calculate final

assembly risk, in addition to guiding the deployment of scarce supplier development teams and resources.

# TABLE OF CONTENTS

	Page
ACKNOWLEDEMENTS.....	iv
ABSTRACT.....	v
LIST OF TABLES.....	ix
LIST OF FIGURES.....	x
CHAPTER	
1. INTRODUCTION.....	1
2. ADVANCED ORDERING STRATEGY.....	4
2.1 Motivation.....	4
2.2 Understanding Advance Ordering Synchronization.....	4
3. ROBUSTNESS OF ADVANCED ORDERING STRATEGY.....	10
3.1 Assumptions.....	11
3.2 Component Profile.....	12
3.3 Modification of Delivery Performance prediction Tool.....	13
3.4 Scenario Runs.....	15
3.4.1 Scenario 1.....	15
3.4.2 Scenario 2.....	19
3.4.3 Scenario 3.....	21
3.5 Summary.....	23
4. IDENTIFYING AND CATEGORIZING SUPPLIER RISK.....	26
4.1 Data Attributes.....	27
4.2 Quantifying Supplier Risk.....	28
4.3 Supplier Prioritization.....	29



4.4 Validation .....	31
4.5 Boosted Poisson Trees Algorithm .....	33
4.5.1 Calculating Supplier Risk Probability .....	35
4.6 Final Assembly Risk .....	35
4.6.1 Sum of Random Variables .....	36
5. CONCLUSION .....	38
BIBLIOGRAPHY .....	41

## LIST OF TABLES

Table	Page
1. Example of 8 components and their corresponding metrics of mean standard deviation and inventory buffer for a 95% service level (Prokle, Beladi and Muriel) .....	8
2. Example of 8 components of the component and assembly service level over time in increments of 5 days (Prokle, Beladi and Muriel) .....	8
3. Scenario 1 and 2 percent improvement in performance measures of AOS vs SLT. This table shows the service level magnitude improvement, Average inventory % reduction, average weeks late reduction and the average coefficient of variation of each scenario ran for 250 iterations .....	24
4. Scenario 3 percent improvement in performance measures of AOS vs SLT. This table shows the service level magnitude improvement, average inventory % reduction, average weeks late reduction and the average coefficient of variation of each scenario ran for 250 iterations ....	25
5. Supplier data aggregation sample with scaled values (not actual data) .....	28
6. Supplier rank by sequential sort method showing Red and Yellow category boundary with 25 percentile group segmentation (not actual data) .....	30
7. Category target variable density with equal 25% split segmentation .....	32
8. Category target variable density with 15%, 20%, 30%, 35% split segmentation (values represent estimates only).....	33

## LIST OF FIGURES

Figure	Page
1. Component weeks late distribution.....	5
2. Existing state with no time buffering synchronization – SLT and.....	7
3. Advanced Ordering Synchronization: 100 <sup>th</sup> percentile time buffering .....	7
5. Calculation of Effective Lead Time $LT_{eff}$ under AOS .....	12
4. Calculation of Effective Lead time $LT_{eff}$ under SLT .....	12
6. 1000-Part System Average Weeks Late Distribution Normalized .....	13
7. Delay implementation diagram.....	14
8. Order behavior under 4 week normal delay (Scenario 1) .....	15
9. 100 <sup>th</sup> Percentile weeks late distribution for the 1500 components simulated .....	16
10. Scenario 1: Final Assembly Average Weeks Late.....	17
11. Scenario 1: Average Inventory Cost.....	17
12. Scenario 1: Average Inventory Cost - SLT.....	19
13. Scenario 1: Average Inventory Cost - AOS.....	19
14. Order behavior with systemic 4 week delay .....	20
15. Scenario 1 and 2 comparison of service level and average inventory .....	21
16. Scenario 3: Percentage of Components Late Performance, SLT vs AOS .....	23
17. Supplier transition matrix from 2013 to 2014 (values are estimates) .....	31
18. Decision tree diagram and its corresponding solution space (Elith, Leathwick and & Hastie).....	34
19. Bill of Material of final assembly in tree structure to calculate final assembly risk from the Poisson rates generating by the BRT algorithm for all components in the assembly (not real data) .....	36

# CHAPTER 1

## INTRODUCTION

There are many facets to analyze when thinking about supply chain risk. Much of the research currently covers supply chain risks from a network disruptions perspective in a holistic fashion. (Tang, 2006) sparks many of the research directions today in Supply Chain Risk Management (SCRM) and suggests four principles: supply management, demand management, product management and information management (Colicchia & Strozzi, 2012). Here we delve into a portion of the network, arguably the most important, supply management. As it is well known that variation in a process propagates downstream and that is why it is important that the inputs to a supply chain be as consistent and risk free as possible. Furthermore, performance of a process' outputs can only be as good as the quality of its inputs (Forker, 1997). In this paper we cover two critical aspects which produce great benefits for the supply chain as a whole when appropriately accounted for. Those two aspects are component on-time delivery and component quality.

There are two events that will stop production before the product at hand starts to be produced, hence affect the bottom-line; that is supplier delivery lateness and significant quality "escapes". Therefore safety inventory and time buffering strategies are a must in many production facilities. But then it becomes a challenge for manufactures to balance these conflicting objectives of inventory cost reduction and customer service levels. As it is commonly known, as inventory levels increase service level rises as well, but on the other hand costs/risk increase with the excess of inventory. Furthermore, dealing with JIT

strategies it can be a challenge to coordinate all components needed for the final assembly of the finished good.

Research has been done by (Beladi, 2014) that allows for 100% certainty that all components will be available for on-time production. In their research it was found that dealing with a system that depends on hundreds of components to show up at specified times (under JIT strategies), planning for the worst case (100<sup>th</sup> percentile in the components average weeks late distribution) for each component results in a dramatic decrease in inventory levels and increases the start production rate up to 100%. We extend this work to answer the question of how robust this system developed by (Beladi, 2014) is. Will it still produce great system performance under unexpected circumstances?

The other parameter in question is quality. Quality can be hard to predict especially with complex components and exhaustive quality check procedures. In many manufacturing processes raw material received from suppliers is assumed to contain no quality non-conformances (QNs) and incoming orders are subject to randomized quality checks with only about 10% of components being checked. In these situations checking all incoming suppliers' shipments is infeasible because of the major resource and cost burden. This is why many OEM manufactures especially in the aerospace industry develop various preventive initiatives and create entire divisions dedicated to supplier quality. The main objectives of these divisions are to oversee supplier operations and uphold policy and regulations. This is of great importance in industries such as aerospace where processes are heavily regulated. Even with these measures in place it is still hard to keep track of all the operations of the hundreds of suppliers distributed around the globe. Also, many of these

suppliers produce several components, which can all have different performance and quality outputs.

For large manufacturing companies, where the number of suppliers are in the mid hundreds and the number of components in the thousands, identifying high risk components is a hard task. Is there a way that a company can identify and mitigate these potential risks by one centralized tool? Many attempts in practice have been made where managers have come together to combine domain knowledge to develop metrics that indicate how a supplier is performing by categorizing the suppliers based off their current performance. One of the major pitfalls is these tools depend on a vast amount of subjective data and the categorization (i.e. suppliers are categorized in Good, Medium and Bad performing groups) tends to be non-informative, especially when trying to allocate resources to implement preventive/corrective action. This paper describes a method and analysis that was developed for a major manufacturer to optimally categorize supplier based on segmented ranking algorithm using readily available data.

## **CHAPTER 2**

### **ADVANCED ORDERING STRATEGY**

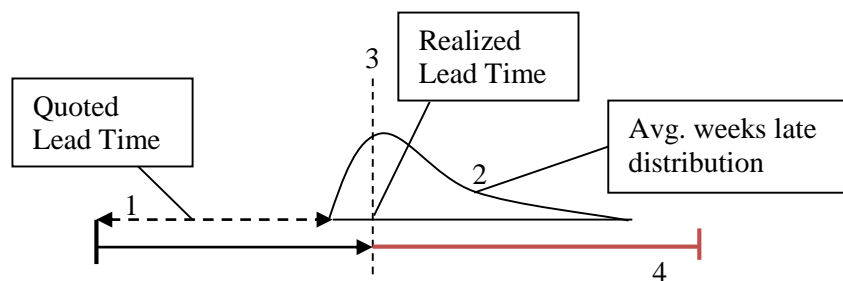
The master's thesis of Faried Beladi at the University of Massachusetts Amherst Department of Mechanical and Industrial Engineering (Beladi, 2014) covered the development of a discrete-event simulation (DES) tool to predict and optimize on time production for an industry component complex manufacturing assemblies. This DES was developed in MATLAB R2011b software and was chosen because of the object oriented programming capability and versatility for extended complex analysis for future work.

#### **2.1 Motivation**

The motivation behind this predictive simulation tool is to provide the ability to explore inventory management strategies to increase on-time production assembly while minimizing inventory. In order to account for component delivery variability, time and physical buffering are commonly used. In the case of an aerospace manufacturer many of the components have high costs and significant variability in delivery lead-times. Practitioners felt inventory buffering would put great financial strain on the company as a whole. It was then decided that discrete-event simulation of the inventory system comprising of the component delivery occurrences and inventory management system would be beneficial to emulate the system behavior in order to find an optimal solution for on-time production improvement and inventory reduction.

#### **2.2 Understanding Advance Ordering Synchronization**

The major driving force of the delivery performance prediction tool is the record of weeks late of each component based on the delivery receipts and Material Requirements Planning (MRP) system's due date taken from a historical six month period. The historical performance of the weeks late metric distributions were created for each component. Viewing Figure 1 displays the foundation of the delivery performance prediction tool and the strategy of on-time delivery described later in this section. What Figure 1 displays is first looking at (1) the Quoted Lead Time (QLT) which was negotiated between the supplier and the Original Equipment Manufacturer (OEM), (2) represents the distribution of weeks late that was observed from over the past six months of that component, (3) represents the realized lead time of when the component actually arrived to the OEM and (4) represents the additional waiting time that the component has to wait for the all the components in the assembly to arrive to start production.



**Figure 1 Component weeks late distribution**

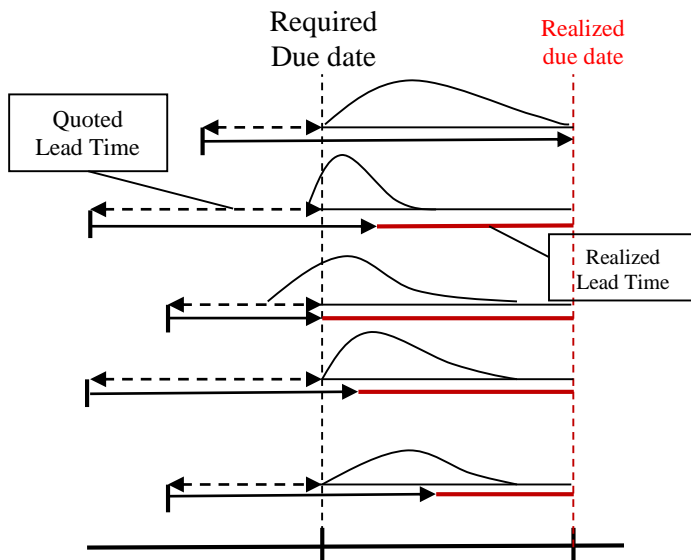
Analyzing the weeks late distributions of the components several strategies were tested using time buffering techniques for the specific assemblies of the manufacturing partner with the goal of achieving at least 95% service level to MRP due date. It was found that the most optimal strategy of time buffering was time buffering all components in the assembly to their respective 100<sup>th</sup> percentile weeks late distribution, hence the Advance



Ordering Synchronization (AOS). A more in depth analysis on the methodology and approach to this finding can be found in (Beladi, 2014) and (Prokle, et al., 2016), but we will provide broad overview in the remainder of this section.

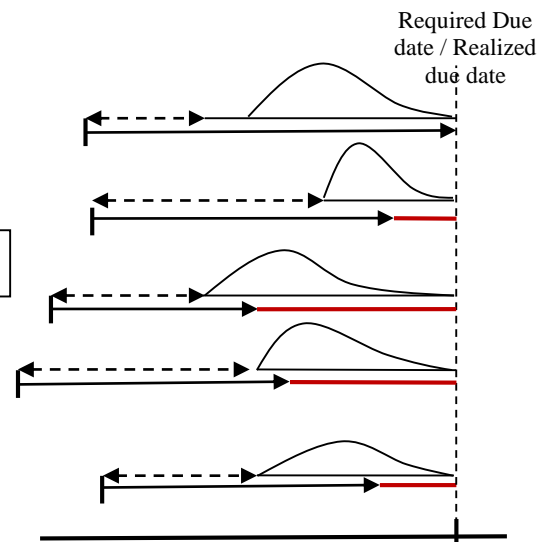
To clearly understand the relationship between the weeks late distribution and on-time production of an assembly we next describe a simple example. Viewing Figure 2 below represents the weeks late distribution of 5 different components in the base case system, SLT. Idealistically, one would want all the components to show up at the same time, which is represented by the black vertical dashed line. But as one can see that a good portion of the probability density of weeks late for each component lands past the required due date. This is where advance ordering is effective. It calculates each components latest possible arrival lateness, and for each component respectively moves up the schedule (or signals to the supplier earlier) of when the component is needed. So comparing Figure 2 and Figure 3 one can see how each components required due date is changed according to their observed arrival lateness. What the comparison of these two figures show is the aligning of all 5 components' 100<sup>th</sup> percentile of the their weeks late distribution to the production start required due date makes sure 100% of the time (theoretically) all 5 components will arrive by the acquired due date.

SLT = No component time buffering



**Figure 2 Existing state with no time buffering synchronization – SLT and**

AOS = All components time buffered to their respective 100<sup>th</sup> percentile weeks late



**Figure 3 Advanced Ordering Synchronization: 100<sup>th</sup> percentile time buffering**

One may say that since each component receiving a time buffer equaled to the 100<sup>th</sup> percentile of their weeks late distribution there will be an occurrence where majority of the components show up early, causing an dramatic of increase inventory, but even though there is a slight probability that this event may occur it is very unlikely due to the high variability in the weeks late distribution of majority of the components, where a good portion of the components will always arrive on their right side of the probability distribution.

Understanding each component's weeks late distribution, time buffering can be applied to a specified percentile that results in a certain on-time arrival probability or service level. Then accounting for the multiplicable factor of hundreds of components' on-time arrival probabilities, one can calculate the overall system on-time production rate.

Referencing (Prokle, et al., 2016), they provide a great an example of necessary inventory buffering to overcome the deterioration of service levels with only 8 components. In the example a production assembly contains 8 components shown in Table 1. and Table 2. Looking at Table 1 it shows the average and standard deviation of the lead time of the component with corresponding inventory buffer. Then Table 2 shows over time (5 day increments) the expected service level of the final assembly accounting for the corresponding inventory buffers from Table 1.

<b>components</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>
<b>Mean</b>	0	0	5	7	25	70	100	100
<b>Standard Deviation</b>	1	10	2	6	4	40	10	40
<b>Inventory Buffer</b>	3.29	32.9	6.58	19.74	13.16	131.59	32.9	131.59

**Table 1 Example of 8 components and their corresponding metrics of mean standard deviation and inventory buffer for a 95% service level (Prokle, et al., 2016)**

<b>Comp. #</b>	<b>Probability of availability after given number of days</b>										
	0	5	10	15	20	25	30	35	40	45	50
<b>1</b>	0.95	1.0	1.0	1.0	1.00	1.00	1.00	1.00	1.00	1.00	1.00
<b>2</b>	0.95	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
<b>3</b>	0.95	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
<b>4</b>	0.95	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
<b>5</b>	0.95	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
<b>6</b>	0.95	0.96	0.97	0.98	0.98	0.99	0.99	0.99	1.00	1.00	1.00
<b>7</b>	0.95	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
<b>8</b>	0.95	0.96	0.97	0.98	0.98	0.99	0.99	0.99	1.00	1.00	1.00
<b>Final Assembly</b>	0.66	0.89	0.93	0.96	0.97	0.98	0.98	0.99	0.99	0.99	1.00

**Table 2 Example of 8 components of the component and assembly service level over time in increments of 5 days (Prokle, et al., 2016)**

Table 2 shows that only considering the inventory buffers for each part the final assembly would result in a service rate of only 66% on the required due date (i.e. day 0), which is shown in the second column from the left. In order to reach an assembly on-time production service level of 95% production would have to wait almost 15 days past the expected start date. Wanting to be more ambitious by reaching a 99% service level, production would have to start 35 days past the expected start date. In practice with assemblies consisting of thousands of components, inventory buffers alone would not be efficient enough to hedge against the variability or would cause a firm tremendous amount of buffer inventory, given the multiplicable factor of combined individual component's service levels as component quantity increases (e.g.  $0.95^{100} = .006$ ) (Prokle, et al., 2016).

## CHAPTER 3

### ROBUSTNESS OF ADVANCED ORDERING STRATEGY

We have shown that the advanced ordering strategy (AOS) is very attractive in reducing inventory and ensuring customer delivery with adding time buffers for all components equaled to their 100<sup>th</sup> percentile weeks late distribution. Throughout this section we will be comparing the current system with no time buffers, SLT and advanced ordering system strategy, AOS. The experiments ran consider the historical component delay distributions, and assume that they are good predictors of the supplier's performance in the future. In practice, however, the weeks late distributions of components from a supplier may shift over time, or experience sporadic severe disruptions. To test the performance of the AOS in such events, we simulate the system in a variety of disruption and distribution shift scenarios. How will the system behave when a component displays a delay far beyond what any historical data analysis could predict? Will AOS still outperform SLT (i.e. the current state of the system)<sup>1</sup>? Will the intuitive belief of time buffering resulting in an overwhelming surge in system inventory pan out?

To answer these questions we simulate and analyze the behavior of a 1500-component system under the SLT versus AOS strategies in the following scenarios:

- *Scenario 1:* single components that are at a particular point in time that are 4, 8, and 12 weeks later than their 100th percentile weeks late,

---

<sup>1</sup> SLT is the system lead-time given from the buyer and supplier contract agreement which is the current state to which the advance ordering system is compared to

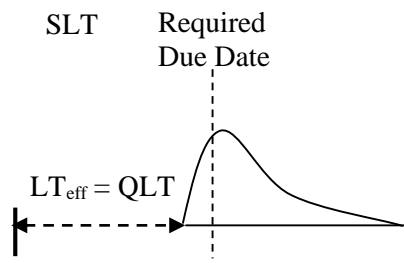
- *Scenario 2*: single components that are 4 weeks late beyond the 100<sup>th</sup> percentile of weeks late consecutively for 4, 8 and 12 weekly orders, and lastly,
- *Scenario 3*: a percentage of components (1%, 3%, 5%, 10%, 15% and 20%) that would be each 4 weeks late beyond their 100<sup>th</sup> percentile to the respective order.

### 3.1 Assumptions

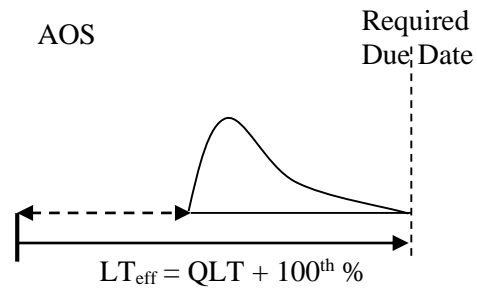
There are a few assumptions for the three scenarios. First we are considering a period of 52 weeks for the simulation length after the warm-up period, and run 250 replications of the 52-week period, in order to estimate annual performance. The same seed is used to produce identical random variables throughout the model to allow for better comparison of the strategies under each scenario. We used a warm up period equal to the maximum effective lead-time over all components in the assembly multiplied by two, which was strategically calibrated to track simulation performance. Effective lead-time,  $LT_{eff}$  is just the lead-time that the simulation uses for a particular component, which has two calculations 1)  $LT_{eff}$  just equals the Quoted Lead Time (QLT)<sup>2</sup> under the SLT case and 2)  $LT_{eff}$  equals the QLT plus 100<sup>th</sup> percentile of the weeks distribution,  $L^{100}$  of the specified component. See Figure 5 and Figure 4 below.

---

<sup>2</sup> The Quoted Lead Time as described previously is the negotiated lead time for orders sent from the OEM to the supplier



**Figure 5 Calculation of Effective Lead time  $LT_{eff}$  under SLT**



**Figure 4 Calculation of Effective Lead Time  $LT_{eff}$  under AOS**

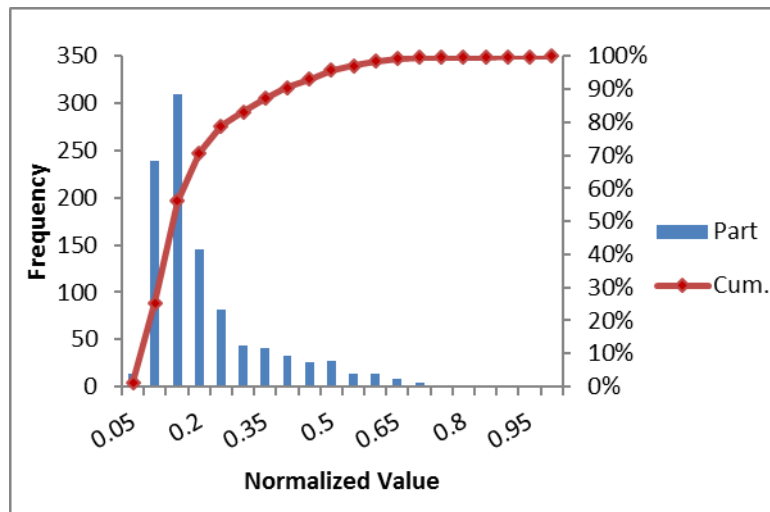
Furthermore, in the simulation the  $LT_{eff}$  is used to time phase all the components to arrive on time for one particular production start date.  $LT_{eff}$  is multiplied by two to make sure the component with the maximum  $LT_{eff}$  is able to have two orders fulfilled before the statistics on performance are taken. To keep simulation runs consistent we took the greater of the maximum effective lead-times (of SLT and AOS cases) and used this value to determine the warm up period for all simulations.

### 3.2 Component Profile

A major factor on the output of all scenarios is component selection. Each component has certain characteristics as follows: Unit cost per assembled product, number of units per assembly, quoted lead-time (QLT), weeks late distribution, including average and standard deviation of weeks late. To visualize component delays across the entirety of components, a heat map has been developed as a two-dimensional graph that displays one point for each component with coordinates equal to its corresponding average and standard deviation of weeks late. The Cartesian distance from the point in the graph to the origin is referred to as the heat map distance (HMD) of the component. The characteristics most significant to the component profile are the unit order cost per assembled product and the heat map

distance. These two characteristics can greatly influence the results of the simulation. The cost per assembled product is simply the cost of one unit multiplied by the units per assembled product. These features should be aware of when analyzing and making sense of results from different scenarios.

To better understand the component make up Figure 6 shows the distribution of the normalized average weeks late across the 1500 components as a histogram. Here we can see that about 80% of the components are probably in a reasonable range of weeks late, but there is a small portion of components that can potentially hold up an entire production line.



**Figure 6 1000-Part System Average Weeks Late Distribution Normalized**

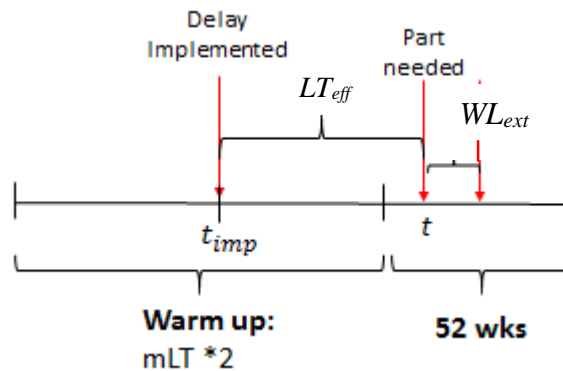
### 3.3 Modification of Delivery Performance prediction Tool

The delivery performance prediction tool was modified in order to perform the analysis of the three scenarios stated above. The basis of the three scenarios is implementing a specified lateness at a certain time  $t$  for a specified component. For the delay implementation there are two related input parameters to consider: 1) the time the delay



should be implemented in the system, and 2) when the component will be needed by the system to fulfill an order. For example in Figure 7 below, the second arrow from the right represents when the component will be needed (i.e., when we want the lateness to occur so that it is fully captured in our 52-week simulation). The first arrow represents when the lateness needs to be implemented, that is, the ordering time  $t_{imp}$  to which the longer supply lead time is assigned. This ordering time takes into account how far in advance orders for that component are placed under the strategy chosen (SLT or AOS). How far in advance the strategy orders a component can be interpreted as the lead time that the strategy assumes for that component, hence  $LT_{eff}$ .

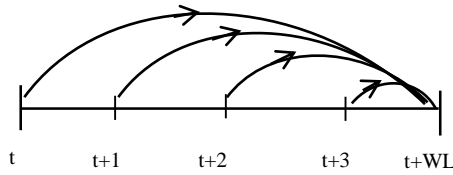
The simple calculation would be:  $t - LT_{eff} = t_{imp}$



**Figure 7 Delay implementation diagram**

We also want to emphasize the assumption briefly discussed in Chapter 2: if an order is late at time  $t$  then all succeeding orders ( $t_i = t_{i-1} + 1$ , where  $i = 1 \dots WL$ ,  $t_0 = t$ ,  $WL =$  weeks late of order delay) will all be delivered at  $t + WL$ . This is to ensure that orders do not cross; a later order cannot arrive earlier. We are assuming that an order scheduled for arrival at

time  $t$  that is late by  $WL$  weeks will also delay orders up to time  $t + WL$ , making all orders being delayed arrive on the same day ( $t + WL$ ). See Figure 8 for visual explanation.



**Figure 8 Order behavior under 4 week normal delay (Scenario 1)**

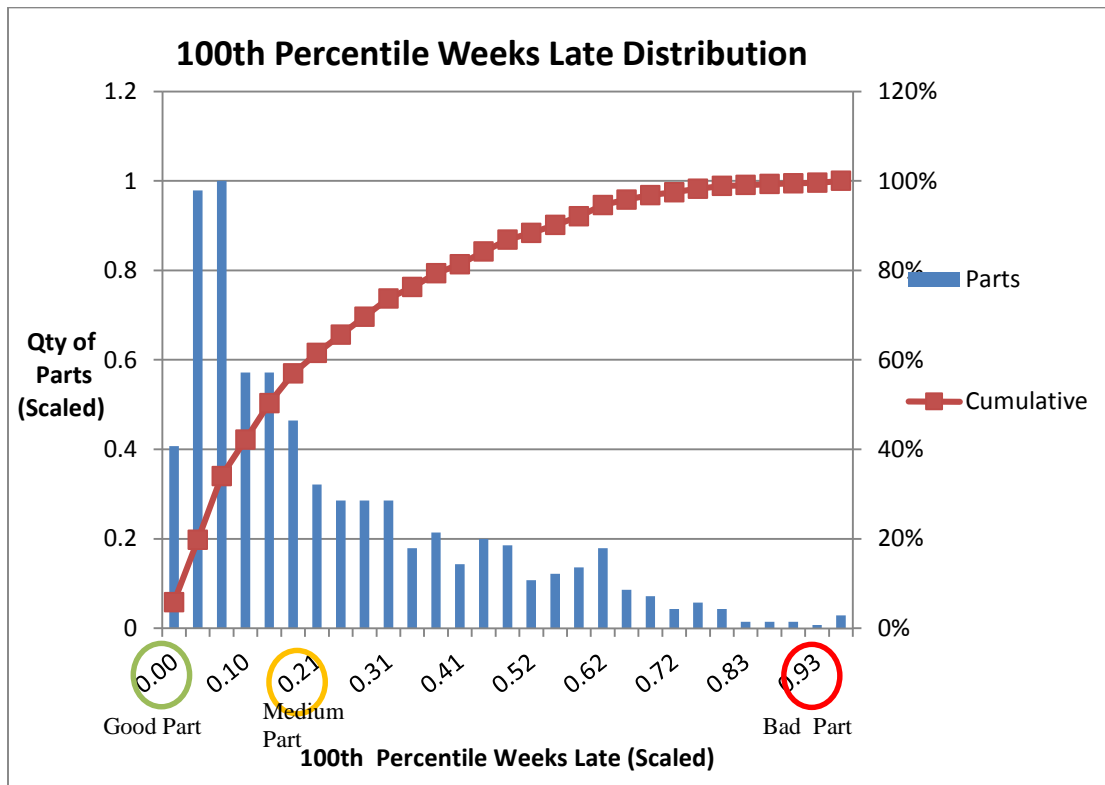
One major difference between the SLT and AOS is the that in the AOS case runs, the  $LT_{eff}$  already accounts for the 100<sup>th</sup> percentile weeks late, where in the SLT as a modeler one has to be conscious that  $LT_{eff}$  is not accounted for and needs to be added to the extreme lateness of the specified component. This allows for an equitable comparison of the two systems types.

### 3.4 Scenario Runs

#### 3.4.1 Scenario 1

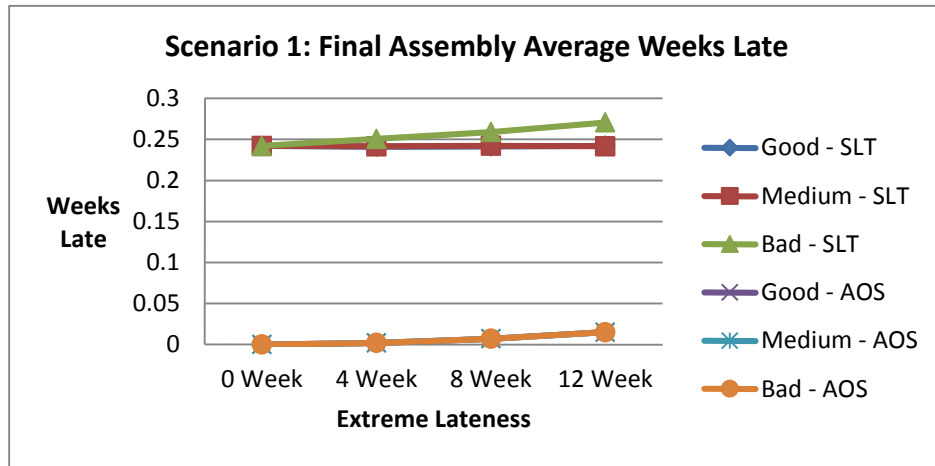
In *Scenario 1* we consider three components that we call “Good Component”, “Medium Component” and “Bad Component.” These components represent the minimum, median and maximum, respectively, of the 100<sup>th</sup> percentile weeks late distribution of components considered for the simulation run. See Figure 9. We assume no components have negative weeks late; a component is either on time or late. Therefore, for our minimum 100<sup>th</sup> percentile weeks late value is equaled to 0.<sup>3</sup>

<sup>3</sup> Negative weeks late represent a part arriving early, before the MRP due date.



**Figure 9 100<sup>th</sup> Percentile weeks late distribution for the 1500 components simulated**

For each type component (i.e. Good, Medium, Bad) a series of simulations were run where an extreme lateness,  $WL_{EXT}$  of 4, 8, or 12 weeks was applied to the specified component. Under each case, we compared how the SLT system performed versus the AOS system. Our results show significant differences. Referring to Figure 10 and Figure 11 one can see significant reductions in final assembly average weeks late and average inventory cost by 97% and 54%, respectively. This validates that the AOS system significantly outperforms SLT even under statistically unforeseen conditions.



**Figure 10 Scenario 1: Final Assembly Average Weeks Late**



**Figure 11 Scenario 1: Average Inventory Cost**

Not surprisingly, the performance of AOS worsens as some components are later than statistically planned for. However, the system performance is still significantly better than that of the current SLT strategy. Now delving into the results deeper, one can see that there are other phenomena occurring when closely looking at the SLT case and more specifically

the Bad Component. These occurrences follow the insight that when the production schedule is being held up because of a few components, inventory costs would decrease if components that usually show up on time (i.e. a “Good component”), actually arrive late. Considering Bad Components arriving even later, this increases the average inventory dramatically by having to hold on to inventory for that extended amount of time, especially if the majority of the components required to start production, have already arrived.

Looking at Figure 12, the SLT case, as the component profile goes from good to bad, the average inventory cost rises increasingly. More specifically there is a tremendous difference in average inventory costs compared to the Good and Bad component cases. Then viewing Figure 13, the AOS case, as the component profile goes from Good to Bad there is very minimal difference, if at all. Not without mentioning the dramatic reduction in overall average inventory. Practitioners implementing AOS will minimize the effect of increasing the average inventory levels with not having any bias towards any good or bad performing components arriving extremely late, hence a robust method of increasing on time delivery, while effectively minimizing inventory costs.

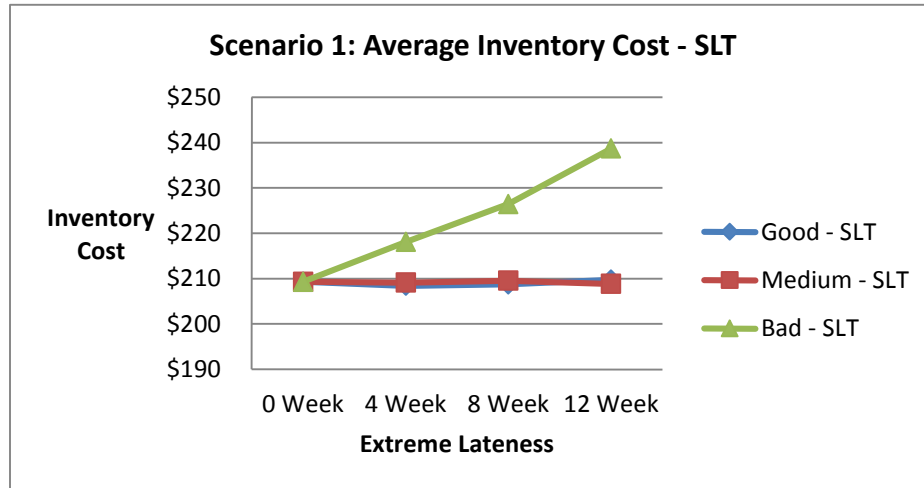


Figure 12 Scenario 1: Average Inventory Cost - SLT

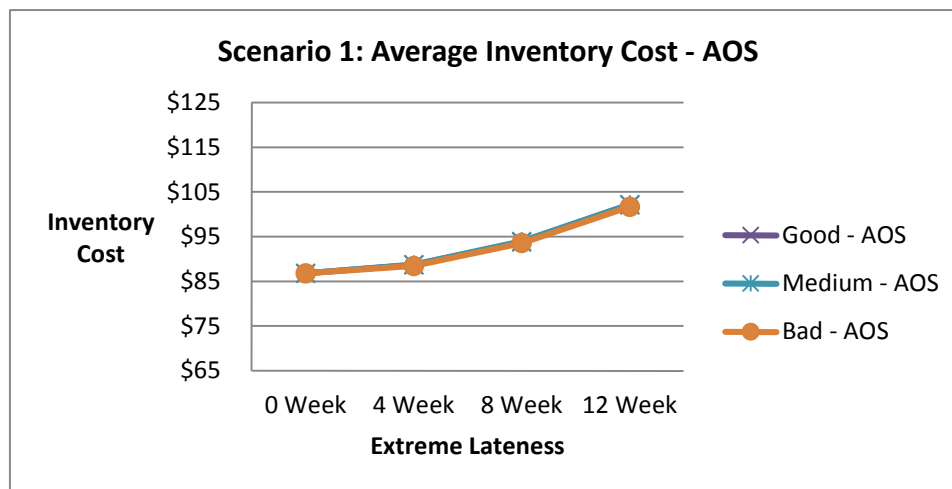
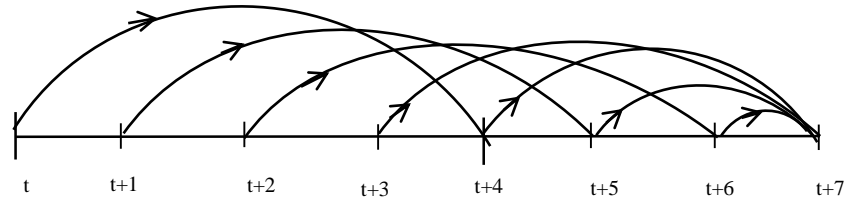


Figure 13 Scenario 1: Average Inventory Cost - AOS

### 3.4.2 Scenario 2

For *Scenario 2* we looked at how the system would behave when components are late on 4, 8 and 12 consecutive orders (which equate to weeks). See Figure 14 below to better understand the order behavior under systematic delay and compare with Figure 7 to see the difference with normal component delay. In this scenario we wanted to gain more insight

on how severe compound lateness would affect the system as a whole, but even more so how would both system strategies compare during these extreme conditions.



**Figure 14 Order behavior with systemic 4 week delay**

Viewing Figure 15 we can see that comparing the results of scenario 1 and 2 the system behaved very similarly with expected decrease in service level and increase in average inventory of about 10% and 12% respectively.

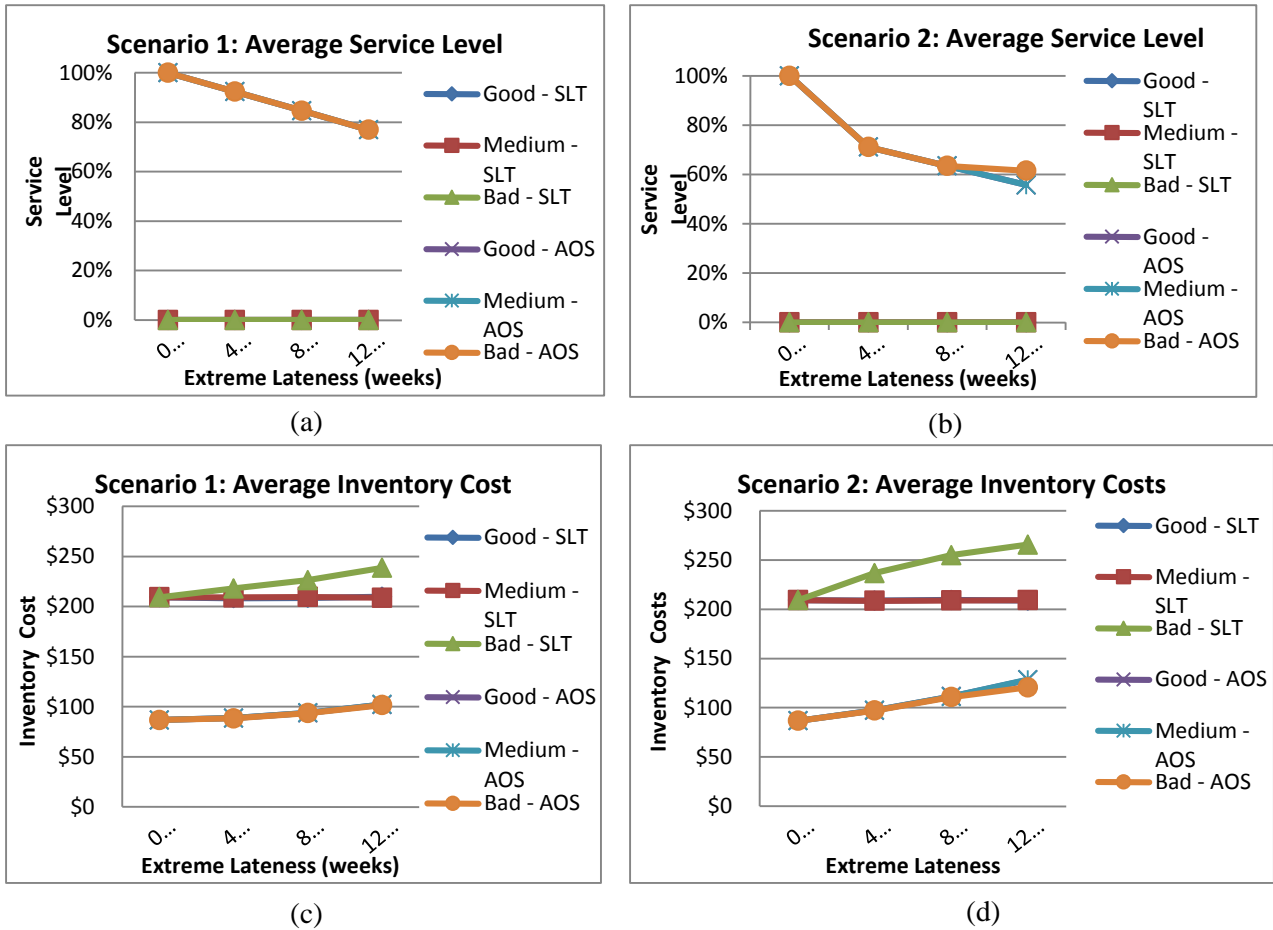


Figure 15 Scenario 1 and 2 comparison of service level and average inventory

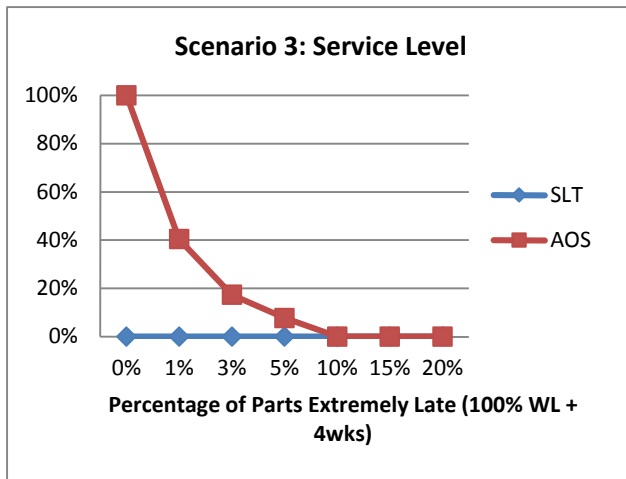
### 3.4.3 Scenario 3

Lastly, in *Scenario 3* we wanted to explore how the system would behave when multiple components were extremely late and to see if there was a critical number of components that would cause the system to fail. For this scenario we made sure to account for components being late on random weeks throughout the simulation. This assumption was that all components considered for the extreme lateness would be given a random time  $t$  between week 1 and week 52 for the start week of extreme lateness (i.e. extreme lateness

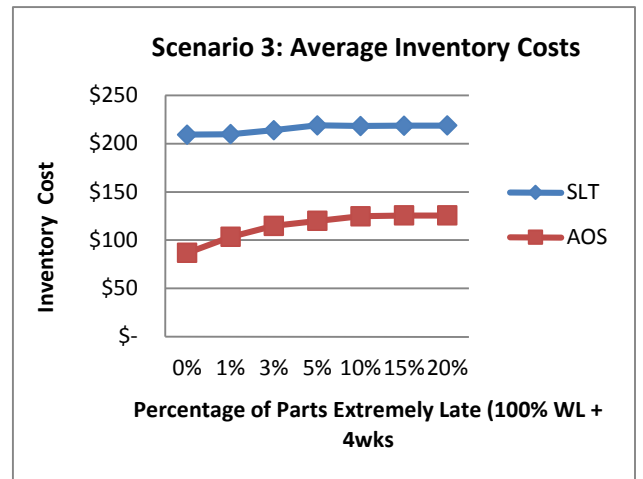


is equaled to four weeks). In a practical sense this would represent a specified number of components randomly arriving extremely late within a 52 week period.

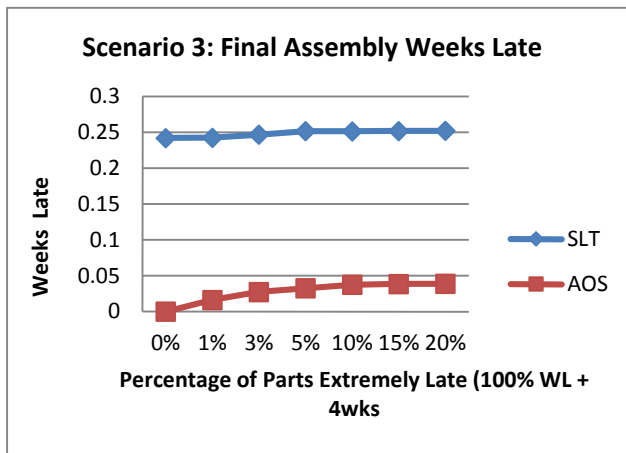
Viewing dFigure 16(a) you can see both systems are performing as expected when the specified percentage of components are extremely late (100<sup>th</sup> percentile weeks late plus four weeks). AOS seems to get increasingly worse off as you increase the percentage of components extremely late and SLT remains at constant service level of 0%. Now looking at dFigure 16(b – d) similar and expected results occur where AOS dramatically outperforms the SLT and their respective curves gradually increase to some asymptotic level. The reasoning behind this limiting behavior is because of the four week extreme lateness constraint that we have imposed on the components. But more specifically dFigure 16c (final assembly average weeks late) concretely shows this aspect and you see how the AOS curve approaches four weeks asymptotically. In all, dFigure 16 shows that the AOS performance considering a percentage of components still outperforms SLT by a great margin.



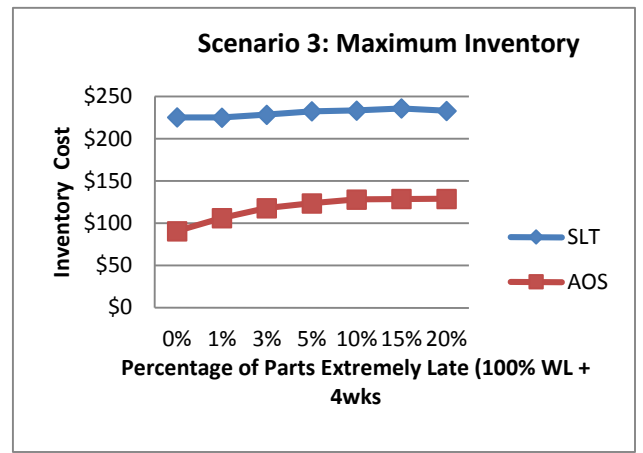
(a)



(b)



(c)



(d)

dFigure 16 Scenario 3: Percentage of Components Late Performance, SLT vs AOS

### 3.5 Summary

We set out to validate that under unforeseen conditions that the Advanced Ordering Strategy (AOS) would perform superior versus the System Lead Time (SLT). We have compared the two during three scenarios: 1) one specified component profile (i.e. Good, Medium and Bad) would experience a lateness equaled to their calculated 100<sup>th</sup> percentile weeks late plus an extreme lateness of 4, 8, and 12 weeks, 2) this scenario is the same as

scenario 1 except that each specified component would experience a systematic lateness equal to their calculated 100<sup>th</sup> percentile weeks late plus an extreme lateness of 4, 8, and 12 weeks, where this systematic lateness would effect not just the one order, but following orders corresponding to the representative extreme lateness, and 3) a percentage of components total components in the assembly (1%, 3%, 5%, 10%, 15%, & 20%) that experience an extreme lateness of 4 weeks.

Furthermore, looking at Table 3 it is easy to see the superiority of the AOS to SLT. As is shown, the average service level improved from 0% to on average 88% and 73% for scenarios 1 and 2 respectively. Additionally, inventory reduction on average was improved by 57% and 52% for scenarios 1, and 2 respectively. Even more so, the average wait time for final assembly completion is reduced nearly 98% and 93% for scenario 1 and 2, respectively.

	AOS vs SLT: % Improvement					
	Scenario 1			Scenario 2		
	Good	Medium	Bad	Good	Medium	Bad
Service Level	*88%	*88%	*88%	*73%	*73%	*74%
Avg Inventory	55.59%	55.59%	58.50%	49.32%	49.30%	57.03%
Avg Wks Late	97.54%	97.54%	97.67%	92.29%	92.29%	93.82%
Avg Coef Var	0.03	0.03	0.02	0.031	0.032	0.022

\*Service level only accounts for the percent magnitude in difference between AOS and SLT

**Table 3. Scenario 1 and 2 percent improvement in performance measures of AOS vs SLT. This table shows the service level magnitude improvement, Average inventory % reduction, average weeks late reduction and the average coefficient of variation of each scenario ran for 250 iterations**

Furthermore, viewing Table 4 is the percent improvement comparing AOS versus SLT. Here we can see we are much better off with the AOS, with much emphasis on average weeks late reduction of 89% across all scenarios. Another aspect to highlight is even though for scenario 3 cases 10%,

15%, and 20% equaling 0% service viewing the average weeks late there is still a great reduction in, which signifies even though components are still arriving late, components are still arriving with a reason amount of time past the required due date

	AOS vs SLT: % Improvement						
	Scenario 3						
	0%	1%	3%	5%	10%	15%	20%
Service Level	*100%	*40%	*17%	*8%	*0%	*0%	*0%
Avg Inventory	59%	51%	46%	45%	43%	43%	43%
Avg Wks Late	100%	93%	89%	87%	85%	85%	85%
Avg Coef Var	0.030	0.028	0.024	0.026	0.026	0.026	0.052

\*Service level only accounts for the percent magnitude in difference between AOS and SLT

**Table 4 Scenario 3 percent improvement in performance measures of AOS vs SLT. This table shows the service level magnitude improvement, average inventory % reduction, average weeks late reduction and the average coefficient of variation of each scenario ran for 250 iterations**

From these scenarios we have concluded that the AOS system does indeed perform better by a great margin and is very robust for different component profiles under several extreme cases. It answers the question of how the system will perform under unforeseen disruptions in the supply base and whether Advanced Ordering System will uphold the uncertainty of the future.

## **CHAPTER 4**

### **IDENTIFYING AND CATEGORIZING SUPPLIER RISK**

Industry wide there has been a push to reduce risk along the supply chain from the result of global expansion. Companies interact with hundreds of suppliers around the globe making it difficult to monitor a supplier's performance and quality. Understanding a supplier's performance and quality allows manufactures to quickly take action to isolate issues and provide the necessary resources to implement corrective action. With our industry partner there has been many attempts to develop a tool that can categorize suppliers in different risk groups to better identify and monitor suppliers to then efficiently allocate resources to improve those suppliers. These tools to date haven't been able to accurately predict a supplier's behavior from year to year, seeing accuracy levels of less than 33%.

Working with the manufacturer the goal was set to accurately identify and categorize supplier risk in order to identify and mitigate quality issues to optimally allocate resources for corrective action of the specified suppliers. These following steps were taken in order to achieve this goal:

- Isolate most relevant data attributes based off domain knowledge and preliminary analysis;
- Identify target variable variable(s);
- Prioritize supplier risk by applying sequential sorting method from most correlated variable to least correlated; and

- Use Boosted Poisson Tree algorithm to find expect values of target value occurrence (Collaboration with UMass Amherst Computer Science Department)
- Use method of sum of independent random variables of the expected values of target value to calculate overall final assembly risk

#### **4.1 Data Attributes**

To isolate the most relevant data attributes we first needed to understand the data that was currently being fed to the company's existing risk tools. After analyzing the risks tools we were able to conclude that a major source of error was from the objectivity of the data attributes. Working with the subject matter experts on the availability of data we were able to cut down our selection to 20 variables that were concluded to be reliable and had minimal human biases. Each variable represents the aggregation of supplier instances. These supplier instances are made up of individual component behavior metrics. Using the industry partner's relational databases made it fairly easy to aggregate all individual component instances by supplier. Then for each supplier there are instances occurring throughout the month from their respective components, which are then aggregated into monthly totals for that supplier creating one instance.

At any given year a supplier will have 12 instances, representing behavior for each month of the year, for each of the designated variables. Viewing Table 5 it shows the aggregated yearly values for supplier with their corresponding variable.

Supplier ID	Variable A	Variable B	Variable C	Variable D	Target variable Value
1	0.99	0.80	1.00	0.55	4
2	0.93	0.86	0.31	0.85	1
3	0.65	0.65	0.26	0.54	0
4	0.45	0.60	0.48	0.71	3
5	0.94	0.62	0.61	0.67	0
6	0.98	0.85	0.51	0.37	0
7	0.37	0.79	0.32	0.59	1
8	0.05	0.91	0.84	0.05	1
9	0.78	0.24	0.28	0.82	2
10	0.41	0.18	0.95	0.21	4
11	1.00	0.93	0.08	0.01	2
12	0.96	0.10	0.44	0.91	2
13	0.21	0.20	0.49	0.01	1
14	0.97	0.01	0.44	0.21	1
15	0.35	0.99	0.55	0.24	4

**Table 5. Supplier data aggregation sample with scaled values (not actual data)**

#### **4.2 Quantifying Supplier Risk**

Supplier risk is used here as a measure of the likelihood of acceptable supplier performance in the near future. Many of our industry partner’s strategies to categorize risk consisted of data attributes that were incorporated into weighted aggregations to output a single supplier score. These scores were then used to rank suppliers. Some of the attributes were the likes of: supplier financial health, manufacturing production scores, number of quality non-conformances over the past year, etc. Through the input of domain knowledge experts, we identified a target variable, which represented the one variable most important concerning supply quality non-conformances. This was an essential step because it allowed analysis of the variables to be mapped to an outcome overtime increasing predictive accuracy. This target variable represented the number of quality non-conformances that

were produced for any given supplier. From this we were able to quantify the suppliers risk based on the number of quality non-conformances a specific supplier produced over a certain time period.

### **4.3 Supplier Prioritization**

From our variables that we have identified (i.e. Variables A, B, C, and D) a sequential sort method was performed to prioritize suppliers creating a ranking system by their relative sorted position. As you can see in Table 6 using Microsoft Excel this sorting scheme was implemented. Looking at Table 6 Variables A, B, C, and D were chosen by performing a correlation analysis of the aforementioned list of 20 variables, with the aggregated values, to the target variable as described in Section 4.2. The highest correlated variables were chosen to be used in our sequential sort method and suppliers were sorted by highest to least correlation coefficient value. From this sort we segmented the prioritized list into four categories in regions of 0-25%, 26-50%, 51-75%, 76-100%, each representing the designated percentile grouping. Here 0% equates to best performing supplier and 100% equates to worst performing supplier. See below for 25 percentile grouping.



Rank	Supplier ID	Variable D	Variable A	Variable C	Variable B
80	11	1.00	0.93	0.08	0.01
79	1	0.99	0.80	1.00	0.55
78	6	0.98	0.85	0.51	0.37
77	14	0.97	0.01	0.44	0.21
76	12	0.96	0.10	0.44	0.91
75	5	0.94	0.62	0.61	0.67
74	2	0.93	0.86	0.31	0.85
73	9	0.78	0.24	0.28	0.82
72	3	0.65	0.65	0.26	0.54
71	4	0.45	0.60	0.48	0.71
70	10	0.41	0.18	0.95	0.21
69	7	0.37	0.79	0.32	0.59
68	15	0.35	0.99	0.55	0.24
67	13	0.21	0.20	0.49	0.01
66	8	0.05	0.91	0.84	0.05

**Table 6 Supplier rank by sequential sort method showing Red and Yellow category boundary with 25 percentile group segmentation (not actual data)**

The next question to answer is how predictive is the sequential sorting method? To answer this question a comparison of supplier categorization in the year 2013 to the year 2014 was performed. For this comparison a supplier rank transition matrix was developed which calculated the amount of suppliers that either stayed in the same 25 percentile category or moved up or down to another 25 percentile category. The transition matrix is made up of four transition states, with 16 different transition type movements. See below.

		Supplier Transition Matrix			
		2014			
2013	Green	67%	27%	4%	2%
	Yellow	12%	65%	15%	8%
	Orange	8%	1%	71%	20%
	Red	0%	2%	6%	92%

**Figure 17 Supplier transition matrix from 2013 to 2014 (values are estimates)**

Looking at the transition matrix in Figure 17 we can see the percentage of suppliers that started in their respective categories in the year 2013 then the resulting categories in 2014.<sup>4</sup> Focusing on the “Red” suppliers, the diagram above shows that about 92% of the suppliers that were categorized as Red in 2013 stayed red in 2014. From the use of the transition matrix it was concluded that these variables can provide predictive insight on the behavior of one supplier from year to year.

#### **4.4 Validation**

To further validate the sequential sorting method the target variable distribution over the four different categories needed to be calculated. This allows us to further prove the method has strong predictive power. Looking below at Table 7 the target variable distribution for each category is shown. As shown, the Red and Orange categories consist of 90% of the target variable instances comparing the 2013 supplier assigned category to the target variable value occurrence in 2014. From this it can be said that this method

---

<sup>4</sup> Real data is not shown; percentages are estimates only.

looking only at the Red suppliers can predict the source of 73% of the potential quality issues that could be mitigated assuming corrective action can solve the issue.

	% of suppliers in each category	% of suppliers containing target variable	% of target variable instances	
Green	25%	5%	3%	
Yellow	25%	10%	5%	
Orange	25%	17%	19%	Orange + Red 92%
Red	25%	68%	73%	

**Table 7 Category target variable density with equal 25% split segmentation**

Arbitrarily splitting the supplier sequential sorting list in to four equal categories leaves much room for improvement to increase the percentage of target variable instances, by reducing Green and Yellow occurrences and increasing the Orange and Red categories. Then arbitrary new quantities of 15%, 20%, 30%, 35% were compared to the previous grouping of 25% for each category. The result here is to move the distribution of suppliers more into categories that represent a higher density of target variable instances. From these new quantities the 2013 Red and Orange categories captured 95% of the 2014 target variable instances showed an improvement of 3% from 92%. See table below for results.

	% of suppliers in each category	% of suppliers containing target variable	% of target variable instances	
Green	15%	5%	2%	
Yellow	20%	10%	3%	
Orange	30%	17%	11%	Orange + Red
Red	35%	68%	84%	95%

**Table 8 Category target variable density with 15%, 20%, 30%, 35% split segmentation (values represent estimates only)**

#### 4.5 Boosted Poisson Trees Algorithm

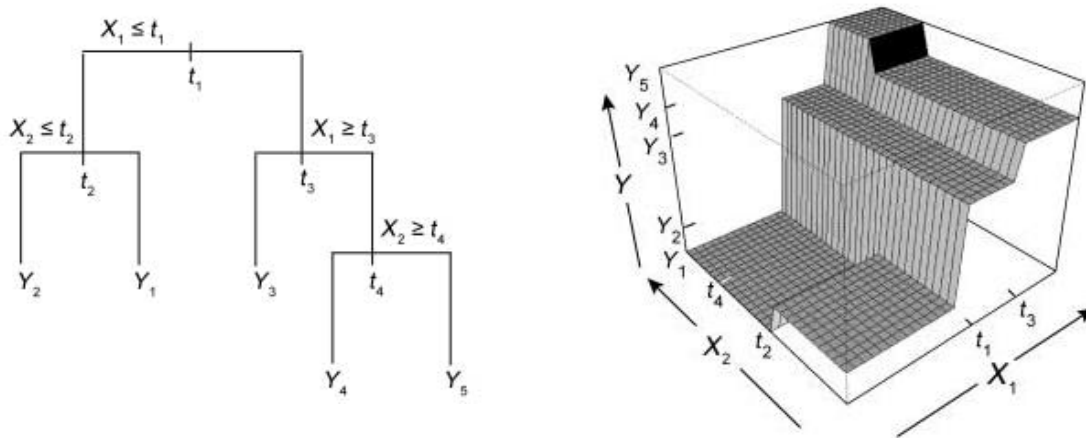
In collaboration with the UMass Amherst Computer Science Department<sup>5</sup> statistical modeling was executed using the Boosted Poisson Trees algorithm. Boosted Poisson Trees come from the family of Boosted Regression Trees (BRT) under the class Machine Learning algorithms (ML). What machine learning algorithms do are instead of being constrained by a certain equation type, these algorithms conform to the patterns in the data and in many instances can have much improved accuracy compared to other prediction methods. In Boosted Regression Trees (BRT) there are two major components: Decision Trees (DT) and Boosting.

First Decision Trees (DT) are great for their simplicity to understand interactions between predictor variables, but are not as accurate than other models (i.e. Support Vector Machines, Neural Networks) (Friedman, Hastie, & Tibshirani, 2001). Decision Trees (DT) are made up of a basic tree structure where each node represents a predictor variable and each leaf nodes represents the response variables. The deeper the tree goes, by adding each additional predictor variable, the more

---

<sup>5</sup> For this application the Boosted Poisson Tree model was developed by Professor David Jensen and his PhD student in the Department of Computer Science at UMass Amherst

accurate the prediction becomes. See Figure 18. In use cases DTs are most popular for classification, where there are set categories that predictions fall in to. However, DTs algorithms can be easily modified for regression analysis. But for most instances DTs tend to be less accurate than other methods. What can be done to increase accuracy in DTs is to grow the tree as deep as possible to catch the many intricacies in the data. But then at the same time this obviously leads to loss in generality and modelers will be in the situation where an increase accuracy can only lead to loss in generality and vis versa.



**Figure 18 Decision tree diagram and its corresponding solution space (Elith, Leathwick, & Hastie, 2008)**

This is where boosting becomes very valuable. In boosting, the DTs do not have to go grow deep at all. In many instances trees are only one or two nodes deep. As one can imagine, these “shallow” trees do not predict with much accuracy, but with boosting accuracy is enhanced greatly. Boosting is considered to be component of the family of algorithms that combine an “ensemble” of weak models to create one strong output. The method that boosting enables is an iterative process

while learning over each data point given from the training data set. How boosting sets itself apart from other methods (i.e. bagging, modeling average) is that it uses an additive process where it builds off the previous iterations by focusing on the magnitude of errors (loss function). With this loss function the model (in regression) tries to create a new tree that minimizes the gradient of the loss function and uses what it learned from the previous trees and calculates the current iteration error (from loss function). In the end a linear combination results of several (most times hundreds) of trees with each tree representing as a term (Elith, Leathwick, & Hastie, 2008).

#### **4.5.1 Calculating Supplier Risk Probability**

The Boosted Regression Tree method that we have described was used to predict the rate that a supplier would produce a non-conforming component. A Poisson loss function was chosen where for this use case was most relevant for the counting aspect of quality non-conformances, therefore producing a rate for each supplier. The model was ran for each supplier to output the rate that a supplier would produce a component non-conformance.

#### **4.6 Final Assembly Risk**

Once the Poisson rates are calculated for each supplier they are then prioritized from greatest to least. From this we created a ranked list for the suppliers that have the greatest probability of producing a non-conformance. From this ranked list of suppliers we now have a systematic way of ranking suppliers based off their probability of producing a non-conformity. In practice, this can be used to guide decision makers to efficiently allocate resources for risk mitigation.

### 4.6.1 Sum of Random Variables

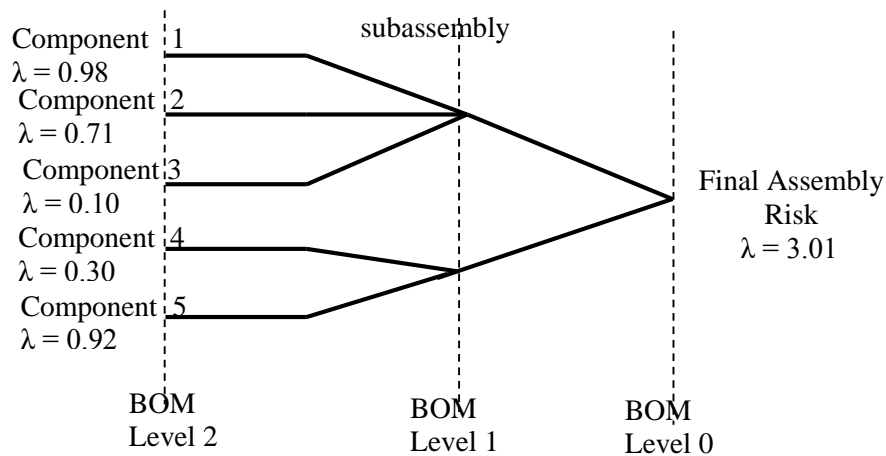
What would be more interesting to see is the overall final assembly risk. When senior management in practice want to get a sense of what the level of risk from the supply base is, it can be difficult question to answer. With the above approach we now have the tools to answer this question. Understanding the basic principles of summing independent variables we can therefore calculate the finished assembly risk using rates obtained from our supplier ranking described in Section 4.5.1

Below is the method of the sum of independent random variables in our case:

$$\lambda_{FA} = \sum_{i=1}^{i=a} c_i ,$$

where  $\lambda_{FA}$  = Final Assembly Risk,  $a$  = number of components in final assembly, and  $c_i$  = component's Poisson rate of quality non-conformance.

Since our supplier ranking model outputs Poisson rates we can easily add up the Poisson rates using the sum of independent random variables method. Summing all the corresponding rates we then have the final assembly total risk. See Figure 19. We can the repeat this step for all different final assembly types and calculate the entire portfolio products risk.



**Figure 19 Bill of Material of final assembly in tree structure to calculate final assembly risk from the Poisson rates generating by the BRT algorithm for all components in the assembly (not real data)**

What Figure 19 represents is the Bill of the Material (BOM) structure of a final assembly consisting of 5 components. BOMs inherently have a tree structure from the parent to child relationship of components to subassemblies to final assemblies. In this example components 1, 2, and 3 come together at the BOM Level 1 to form a subassembly and also components 4 and 5 follow the same process. Then at BOM Level 0 both subassembly join to form the final assembly. Now looking at the risk of the final assembly, because each component has a Poisson rate we can easily sum up the rates, which equaled to 3.01 for the expected final assembly risk, which is also equaled to the expected non-conformance quantity of the entire build of the final assembly.

In all, we establish a simple method that creates a great baseline for any detailed analysis when attempting to predict supplier non-conformances. We then explain the Boosted Poisson Tree algorithm that is more advanced, but very effective in this application. Then we show that using the output from the BPT model, one can easily find the aggregate risk for a system, in this case for a manufactured component assembly.



## CHAPTER 5

### CONCLUSION

When looking at Supply Chain Management one should take a serious look at the supply base. Based on the principle that variation propagates down the supply chain and the final product is only as good as its inputs (Forker, 1997) it is an easy decision to focus on the supply chain inputs. In this paper we concentrated on two inputs: 1) on-time delivery and 2) quality.

We have described the innovated discrete-event simulation tool used to predict supplier performance by the advanced ordering strategy (AOS) developed by (Beladi, 2014). Here each component's average weeks late distribution drives the simulation where time buffering strategies were run in order to achieve a MRP service level of 95% or greater. It was then shown that because of the multiplicative factor of the probabilities of the component lateness in the final assembly, that only a minimal number of components arriving late to production would be needed to quickly diminish the overall assembly MRP service level below the goal 95%, hence the optimal solution was to time buffer all components to their respective 100<sup>th</sup> percentile weeks late distribution.

Then the extension of the delivery performance prediction tool was presented in order to stress test the recommended strategy of buffering all components. We showed three scenarios: 1) single components at a particular point in time that are 4, 8, and 12 weeks later than their 100<sup>th</sup> percentile weeks late, 2) single components that are at a particular point in time 4, 8, and 12 weeks later than their 100<sup>th</sup> percentile weeks late, 3) a percentage of components (1%, 3%, 5%, 10%, 15% and 20%) that would be each 4 weeks late beyond

their 100<sup>th</sup> percentile to the respective order. From these scenarios we prove that the AOS strategy is by far superior to the existing system under the previously mentioned extreme cases. Results show that the average service level improvement comparing the AOS to the existing system is 79%, 64%, and 14% for scenarios 1, 2 and 3 respectively. Additionally, the average inventory reduction was 54%, 48% and 43% for scenarios 1, 2, and 3 respectively.

Lastly, we present a method that categorizes suppliers based off a relative risk level that is driven by the quality of components that the supplier delivers. Several data attributes were reviewed where four variables were ultimately chosen based off having the highest correlation coefficient, when paired with the target variable. Then the sequential sorting method was applied in order of the highest correlated variable to the least. We next tried two arbitrary segmentation methods: 1) four equal 25% splits of the sorted supplier list into categories of Green, Yellow, Orange and Red, where Green being the best performing supplier and Red being the worst; 2) splits of 15%, 20%, 30%, 35% categorized by Green, Yellow, Orange and Red respectively. To see if these methods had predictive power we developed a transition matrix that tracked the categorical movement of the suppliers from the year 2013 and 2014, which show great promise that the analysis was heading in the right direction showing that 92% of the suppliers that were categorized as Red in the year 2013 stayed Red in the following year in 2014.

Next to validate if indeed this method was predictive in quality non-conformances the target variable density was matched to their respective categorized suppliers. Results showed that using segmentation method (1) had 92% of the target variable density resulted from suppliers categorized as Orange and Red. Then even further improvement was made

when implementing segmentation method (2) resulting in 95% of the target variable captured in the Orange and Red categories.

Then finally we incorporated techniques normally used in the data scientist's tool kit in collaboration with UMass Amherst Department of Computer Science; that is the Boosted Regression Trees algorithm where the innovated feature of boosting provides great improved accuracy of the relatively simplistic Decision Tree algorithm. We found using this method a suppliers predicted non-conformance occurrence was enhanced, which can provide a prescriptive source for risk mitigation efficiently allocating resources. This allowed us to output Poisson rates of the predicted quantity of non-conformances in order calculate the final assembly risk. Once having all the Poisson rates for each component in the Bill of Material (i.e. BOM, which incorporates all the components required for completion of the final assembly and their parent-child dependencies) we were able to apply the method of random sum of independent variables and to simply sum up all the components in BOM to calculate the final assemblies overall risk (i.e. expected quantity of non-conformances).

## BIBLIOGRAPHY

- Beladi, F. (2014). *Delivery performance prediction tool for complex assembly systems*. Amherst: University of Massachusetts Amherst.
- Colicchia, C., & Strozzi, F. (2012). Supply Chain Risk Management: A New Methodology for a systematic Literature Review. *Supply Chain Management: An International Journal*, 403-418.
- D.D., W., & J.G., W. (1976). Material Requirement under uncertainty. *Decision Science*, 7, 595-606.
- Elith, J., Leathwick, J. R., & Hastie, T. (2008). A Working Guide to Boosted Regression Trees. *Journal of Animal Ecology*, 802-813.
- Forker, L. (1997). Factors affecting supplier quality performance. *15*, 243-269.
- Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The Elements of Statistical Learning*. Berlin: Springer.
- Gallien, J., & Wein, L. (2001). A Simple and Effective Component Procurement Policy for Stochastic Assembly Systems. *Queueing Systems*, 221-248.
- Prokle, M., Beladi, F., Muriel, A., Subbu, R., Greene, C., & Heslin, J. (2016). Safety Stock Optimization for Assembled Systems with Long and Highly Variable Component Lead Times.
- Tang, C. (2006). Perspectives in supply chain risk management. *International Journal of Production Economics*, 103(2), 41-88.