October 2021

Robustness of Supply Chain Synchronization Strategies

Andrew Frere

Recommended Citation

Frere, Andrew, "Robustness of Supply Chain Synchronization Strategies" (2021). *Masters Theses*. 1143. https://doi.org/10.7275/24575427.0 https://scholarworks.umass.edu/masters_theses_2/1143

Robustness of Supply Chain Synchronization Strategies

Andrew Frere

ROBUSTNESS OF SUPPLY CHAIN SYNCHRONIZATION STRATEGIES

A Thesis Presented

by

AJ FRERE

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

MASTER OF SCIENCE IN INDUSTRIAL ENGINEERING AND OPERATIONS RESEARCH

September 2021

Mechanical and Industrial Engineering

ROBUSTNESS OF SUPPLY CHAIN SYNCHRONIZATION STRATEGIES

A Thesis Presented

by

AJ FRERE

Approved as to style and content by:

____
Ana Muriel, Chair

____
Hari Balasubramanian, Member

____
Senay Solak, Member

____
Sundar Krishnamurty, Department Head
Mechanical and Industrial Engineering

DEDICATION

To my wife, for coming along with me on this journey.

ACKNOWLEDGEMENTS

Thank you to the members of my thesis committee, Hari Balasubramanian and Senay Solak, for lending their guidance and knowledge to this work.

Thank you as well to Raj Subbu and Graeme Johnson for supporting me in this effort.

Thank you to all my friends and colleagues for putting up with me during all of this.

And most of all, thank you to my advisor Ana Muriel, without whom I would not be where I am, in more ways than one.

ABSTRACT

ROBUSTNESS OF SUPPLY CHAIN SYNCHRONIZATION STRATEGIES

SEPTEMBER 2021

AJ FRERE

B.S., UNIVERSITY OF MASSACHUSETTS AMHERST

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

Directed by: Professor Ana Muriel

Modern manufacturing systems dealing with complex assemblies with large numbers of parts present particular challenges in the realm of supply chain management. Complex assemblies, such as those found in aerospace and automobile manufacturing, require thousands of parts to come together at the right time for final assembly. The large number of parts, often coming from hundreds of suppliers, combined with unreliable delivery times and high cost of many of these components can lead to incredibly high inventory costs and assembly delays. Typically, variable delays in part delivery are compensated for by either keeping a buffer of safety stock or a time buffer on the planned lead time of a component. In this thesis, we study the performance of various buffering strategies across a large range of practical scenarios in an effort to identify dominant, robust strategies and how their performance is impacted by the various parameters that define the system. The major conclusion is that aggressive part buffering consistently results in not only better delivery performance but also significant inventory reduction across all settings for assemblies with more than 500 parts.

TABLE OF CONTENTS

Page

ABSTRACT vi
BACKGROUND 1
BUFFERING STRATEGIES 8
2.1 Stochastic Optimization Model 10
COMPUTATIONAL STUDY 14
3.1 Simulation Design 14
3.1.1 Step 1 15
3.1.2 Sampling 15
3.1.3 Step 2 17
3.1.4 Step 3 17
3.2 Simulation Parameters 18
3.3 Simulation Strategy 19
3.3.1 Design of Experiments: Average Performance Analysis 20
3.3.2 Individual Case Analysis: Comparison of Cost and Delivery Distributions 21
COMPUTATIONAL RESULTS 23
4.1 Average Performance Results 23
4.2 Comparison of Cost and Delay Distributions 28
CONCLUSION 38
FURTHER WORK 40
BIBLIOGRAPHY 42
APPENDIX A: THE IMPACT OF CLIPPED DISTRIBUTIONS 43

CHAPTER I

BACKGROUND

This work builds upon previous research conducted at the Supply Chain Management Lab at UMass Amherst, which is described in Beladi (2014), Prokle (2017), Muriel et al. (2018). The objective is to generalize the study, strengthen the conclusions, and augment our understanding of assembly systems under variable component supply lead times in a wide variety of practical cases.

The previous research established a counter-intuitive principle: in situations where a large number of individual components are needed to support a single assembly an aggressive time buffering strategy to mitigate delivery variability provides both reduced assembly lateness and reduced inventory holding cost. The fundamental basis for this argument is the statistical fact that as the number of components increases, the possibility of an exceptionally late arrival (delivery outlier) becomes more and more certain, while the ensuing inventory cost rises sharply as all components wait for the pacing one.

As an illustrative example, consider an assembly system of just 8 components, and a seemingly aggressive buffer strategy that orders each component to achieve a 95% service level; that is, components are ordered early, allowing a buffer time based on their delay distribution that will ensure the part arrives before the targeted assembly time with probability $1-\alpha=0.95$ (see Figure 1)

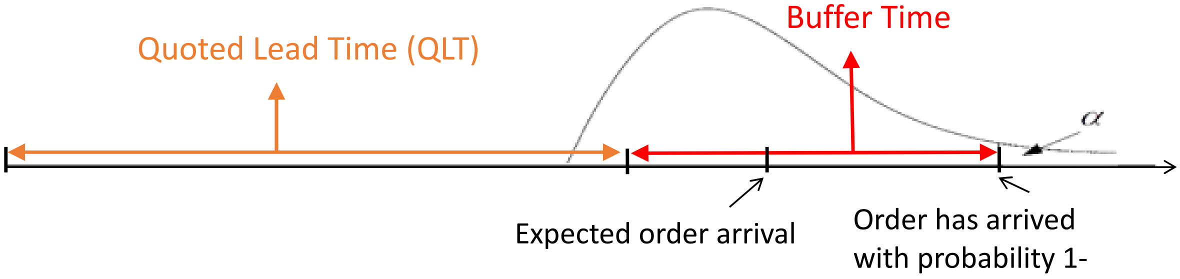
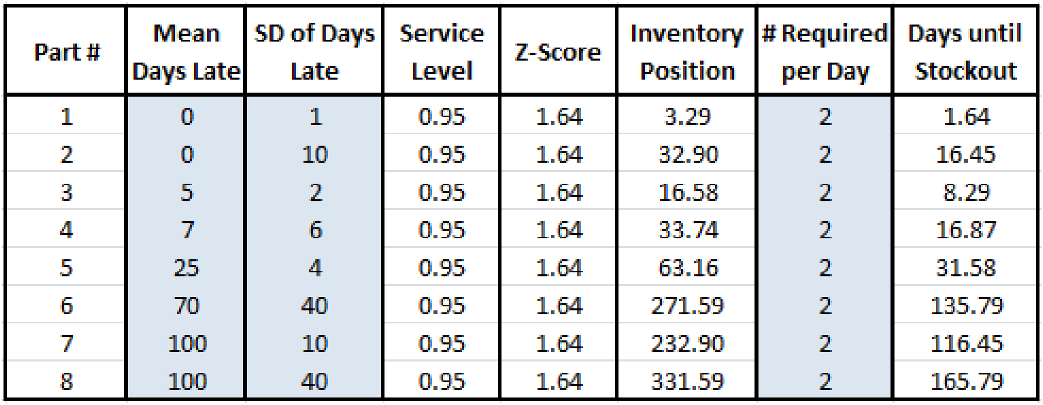
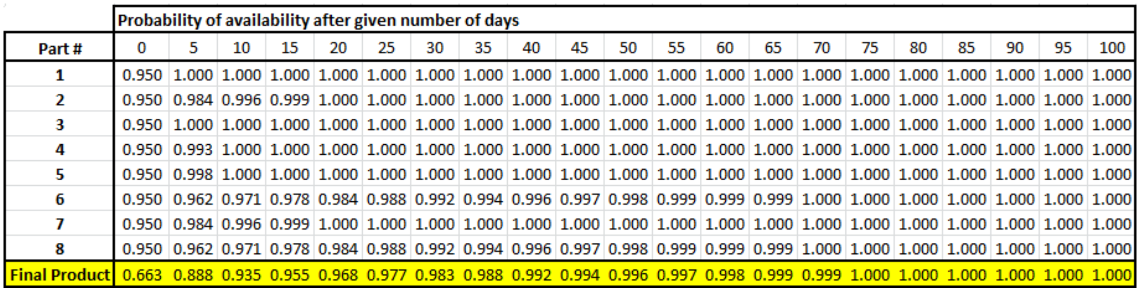


Figure 1 – Arrival distribution of a component with a variable lead time



Part #	Mean Days Late	SD of Days Late	Service Level	Z-Score	Inventory Position	# Required per Day	Days until Stockout
1	0	1	0.95	1.64	3.29	2	1.64
2	0	10	0.95	1.64	32.90	2	16.45
3	5	2	0.95	1.64	16.58	2	8.29
4	7	6	0.95	1.64	33.74	2	16.87
5	25	4	0.95	1.64	63.16	2	31.58
6	70	40	0.95	1.64	271.59	2	135.79
7	100	10	0.95	1.64	232.90	2	116.45
8	100	40	0.95	1.64	331.59	2	165.79

Figure 2 – Lateness parameters for a selection of components



	Probability of availability after given number of days																				
Part #	0	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
1	0.950	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
2	0.950	0.984	0.996	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
3	0.950	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
4	0.950	0.993	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
5	0.950	0.998	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
6	0.950	0.962	0.971	0.978	0.984	0.988	0.992	0.994	0.996	0.997	0.998	0.999	0.999	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000
7	0.950	0.984	0.996	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
8	0.950	0.962	0.971	0.978	0.984	0.988	0.992	0.994	0.996	0.997	0.998	0.999	0.999	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000
Final Product	0.663	0.888	0.935	0.955	0.968	0.977	0.983	0.988	0.992	0.994	0.996	0.997	0.998	0.999	0.999	1.000	1.000	1.000	1.000	1.000	1.000

Figure 3 – Probability of being able to begin work on the final assembly if components are ordered to the 95th percentile of their lateness distributions

Figure 2 provides the lateness parameters for the eight components – note the variation in the mean and standard deviation of the number of days late expected for each component. Figure 3 shows the effects of the compounded delays. Though each component is ordered to what appears to be a high level of service (buffered so that there is a 95% chance the component will arrive before the planned assembly time), the compounded effect of these probabilities is such there is only a 66% chance of being able to begin work on the final assembly on time. The assembly will be nearly fifteen days late for the targeted 95% service level. In other words, there is still about a 5% chance that the assembly will be more than 15 days late, despite the seemingly aggressive buffers. This is because of the compounding of probabilities – across n components the real probability of all components being available is SL^n, where $SL=1-\alpha$ here indicates the targeted service level of each component. As n rises, so does the probability of the assembly being on-time fall exponentially. With 1000 components buffered to a 99% service level, the probability of on-time assembly is $P = 0.99^{1000}$, which is effectively 0. Even a 99.9% service level provides only a 36.7% chance of on-time assembly when spread across 1000 components.

While the above math demonstrates that an aggressive time buffering strategy is necessary to meet adequate on-time performance for assembly, the other argument, that the aggressive buffer also reduces inventory holding cost, is more counter-intuitive and thus requires further investigation.

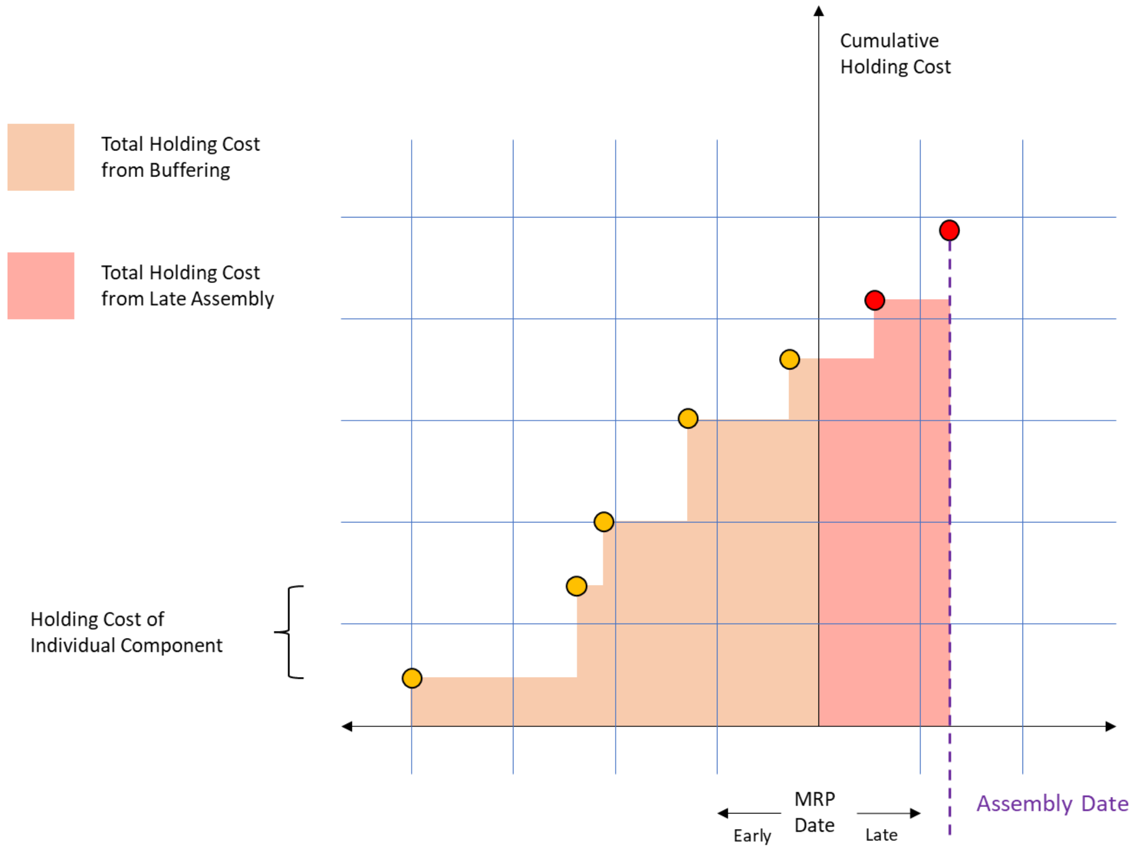


Figure 4 – The delivery performance of the components for a seven-piece assembly. The horizontal axis shows delivery performance in time (the further to the left, the earlier the part was, with the target MRP date marked). The vertical axis shows the inventory holding cost per unit of time for each component (a larger vertical gap between parts representing a higher holding cost for the next part)

Consider the example provided in Figure 4. In this example we have a single delivery of seven components. Five of the components arrived before the target date, while two components had what proved to be an insufficient time buffer for this particular scenario and arrived late. The total holding cost contributed by an individual component is equal to its holding cost per unit of time multiplied by the amount of time that component is waiting for assembly to begin. Note that this waiting period is not equal to the amount of time the component was early by – each part must also wait for all other parts before assembly can begin. Thus, the assembly date, marked in purple in this example, is determined by *the lateness of the latest component.* This is the key point. The shaded regions visually show what the holding costs incurred in this example are. The orange region is the holding cost incurred by buffering; this holding cost is driven by the early arrival of several parts. The red area shows the holding cost contributed by the two parts that were late. The total shaded region represents the total holding cost for this delivery set. Note that while early components only increase the total holding cost by an amount proportional to that component's holding cost, the last component's contribution is proportional to the sum of the holding costs for *all other components.* In other words, every additional day the last component is late means we must hold all other components for an additional day. In comparison, if the earliest component is a single day earlier it only increases total holding cost by its own holding cost value. This demonstrates that minimizing the lateness of the latest component is far more significant to the total cost than curtailing early arrivals. It is the lateness of the worst performing component that drives the cost of the system.

The examples above clearly illustrate two effects:

- *The impact of compounding probabilities:* as the number of components in an assembly increases, so too does the distribution of the lateness of the latest component. Restated, the probability that a component will arrive toward the tail of its distribution (the probability that at least one component will be a severe outlier) increases as the number of components increases.
- *The increasing cost of delay*: while the cost of buffering a particular component remains fixed, proportional to its own cost, the cost of a delay rises as the number of components grows since ALL of the parts will need to be carried in inventory until the last pacing part arrives and the assembly can be completed.

It can then be concluded that as the number of components in an assembly increases it is beneficial in both cost and assembly delay performance to aggressively buffer the delivery lead time of each component against an increasingly high probability of lateness and associated inventory cost.

These examples provide a mathematical and conceptual understanding of the factors that determine the cost and delivery performance of assembly systems. Let's now quantify their impact and identify attractive buffering strategies to mitigate the pervasive negative effect of unreliable delivery times across a wide-range of industry settings. The original research was done in conjunction with an industry partner in the aerospace sector and demonstrated the remarkable benefits of full time buffers (advance ordering components to the worst observed delivery time for each component). Compare the aerospace supply chain with the automotive supply chain. Both systems involve large numbers of individual components coming together for a single final assembly, but there the similarities largely end. The aerospace supply chain has a relatively small number of players working with specialized processes and low demand, in contrast to the automotive sector. As such, while the aggressive buffering strategy was found to be effective for the case of the aerospace partner, demonstrating the robustness of this strategy in other settings is of high importance. This leads us to the two major research questions we seek to answer in this thesis:

1) Is aggressive (full) buffering attractive in other supply chain settings beyond the case study analyzed in the previous work? The goal is to identify the supply chain characteristics under which heavy time buffering of components results in improved inventory cost performance.

2) What is the impact of salient supply chain parameters on the cost and delivery performance of various buffering strategies? In particular, we need to understand the effect of the major parameters that define the supply system: the number of components that make up the assembly, the relative number of components with unreliable delivery times, their delay distribution, the relative cost of these components, as well as the information available to the firm when designing and implementing their buffering strategy.

CHAPTER II

BUFFERING STRATEGIES

To answer the above research questions, we designed a simulation to evaluate the effectiveness of aggressive buffering in different circumstances. This requires defining several buffering strategies, some more aggressive than others, that can be evaluated against each other.

The first strategy we wanted to consider was a no-buffering strategy to serve as a baseline for the effectiveness of different strategies.

We also wanted to consider likely strategies that industry might use. For this we were interested in relatively simple strategies that would be easy to implement with little computational requirements. Buffering components according to the mean of their observed lateness seemed the most obvious as well as highly defensible in industry. However, the lateness distributions frequently have a high skew affecting the mean results, so we also decided to examine a buffering strategy utilizing the median of the observed component lateness as perhaps a better representation of the distribution.

The focus of this research is on the effectiveness of aggressive buffering strategies, so the remaining strategies shown would need to be more aggressive than the mean or median so that an adequate comparison could be performed. The prior research performed by Muriel et al. (2018) advocated for a “full buffering” strategy based on the observed deliveries. In other words, they found that observing recent component delivery data and buffering the delivery of new components by the maximum observed lateness for that component was an effective strategy. This is one of the key points we were interested in testing, so we included this as one of the strategies. This full buffer strategy does not take into account anything except component lateness; it is possible that based on the specific lateness distribution and the particular holding cost of each component that this strategy might not be the optimal approach. Thus, a stochastic optimization model was applied to historic observed data to find the ideal component buffering for each component for that observed set. This buffering strategy would be optimized for the observed data but might not be optimal for the future deliveries based on the random nature of the lateness distributions.

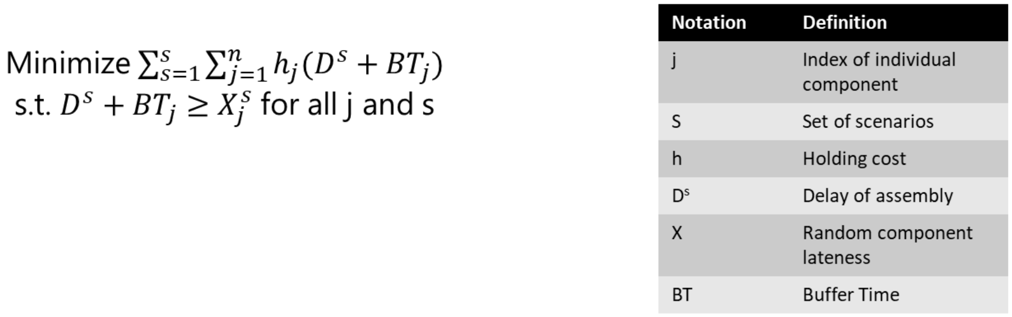
We were also interested in different means of augmenting the prior observed historical deliveries. In an industrial setting different components would be needed in different demand quantities; while some components are ordered in dozens or hundreds per day, some components are only ordered weekly, or even more rarely depending on the specific industry and position in the product life cycle. As such, getting the necessary quantity of delivery data to have confidence in an accurate representation of the underlying distribution could require months, or potentially even years, worth of delivery data. This raises the concern that the delivery distribution may have changed as the performance of the supply chain changes over time. To combat this, we wanted to evaluate two different means of sampling the observed historical data to hopefully provide improved buffering guidance.

The first means of generating new data sampling we chose was to fit a lognormal distribution onto the observed data. From this fitted distribution we could simulate a number of new deliveries and build an optimal buffering strategy for each component from the stochastic optimization model. A different approach was required for the “full buffer” strategy, however, as the lognormal distribution being used has no maximum value. On testing we determined that 3σ of the fitted distribution was an effective strategy in most cases, though we will share results below of testing the effectiveness of different distribution values.

While we felt that a fitted distribution would provide a reasonable platform from which to generate samples, there was concern over how accurate the fitting would be, particularly in cases with a low number of observed deliveries. To that end, we also utilized a second means of data resampling: taking random samples for each component from that component's observed delivery history. This method allows us to recombine existing data to create new delivery profiles without creating new data. As an example, a sampled delivery may have the lateness from component 1 being the lateness of its first delivery in the observed historical data, while component 2 is as late as it was on the eighth delivery. This creates new deliveries in terms of the lateness of each individual component without creating new lateness data. For this sampling methodology we applied the stochastic optimization to find an optimal method based on the resampling, but a full buffer strategy would be the same as for only the observed data, as no new data was generated.

2.1 Stochastic Optimization Model

Three of the strategies we evaluated included finding an optimized strategy according to the observed and sample data. This optimized time buffering strategy was determined according to a linear optimization (Figure 5).



Minimize $\sum_{s=1}^{s}\sum_{j=1}^{n} h_j(D^s + BT_j)$
s.t. $D^s + BT_j \geq X_j^s$ for all j and s

Notation	Definition
j	Index of individual component
S	Set of scenarios
h	Holding cost
D^s	Delay of assembly
X	Random component lateness
BT	Buffer Time

Figure 5 – The linear optimization model used to find an "optimal" time buffering strategy

This model finds the minimum costs by varying the buffering time for each component and the lateness of the assembly (which is determined by the lateness of the latest component). The derivation of this optimization is as follows:

$$Holding\ Cost\ of\ Component\ j = \ h_j(holding\ time\ of\ j)$$

$$Holding\ Time\ of\ Component\ j = \ D^s - arrival\ time\ of\ j$$

$$Arrival\ Time\ of\ Component\ j = \ X_j^s - \ BT_j$$

Substitution and distribution allows us to rearrange the above into:

$$Holding\ Cost\ of\ Component\ j = \ h_j(D^s - X_j^s + BT_j)$$

Note that X_j is a constant – it is the specific lateness being observed for a specific delivery determined by that component's lateness distribution according to Step 1 of the simulation (Chapter 3.1.1). Since this is an observed value that cannot be adjusted by the optimization, its presence is irrelevant. While the specific holding cost would be affected, the D^s and BT_j values that produce the minimum cost would be the same regardless of if X_j was considered or not. Since the goal of the optimization is to find the optimal values of BT_j, X_j can be safely ignored, yielding the component holding cost formula utilized by the above objective function. Note as well that the buffer time is considered only in relation to the component j regardless of which scenario is being run, as this methodology must produce a single buffer time that can be applied to a component for all future deliveries.

The derivation of the constraints is presented below:

$$D^s \geq X_j^s - BT_j$$

This inequality establishes that for a given scenario S the assembly time must be equal to or later than the arrival time of all components – in other words, you cannot begin an assembly until every single component has arrived. Simple rearrangement of the inequality produces the constraints in the optimization model. This rearrangement was necessary to support the data structure required by the Gurobi optimizer utilized by the simulation.

The solution to this optimization model will provide the buffer time for each component that will provide the minimum cost across all the observed deliveries generated by the first step of the simulation (Chapter 3.1.1). Thus, these buffer times are considered "optimized", though in the evaluation phase other strategies may outperform the "optimal" strategy due to random variation.

2.2 Selection of Buffering Strategies for Further Study

From the above, we settled on the following eight strategies for buffering component lateness:

1. No buffer: no mitigation is made for part lateness. Parts arrive when they arrive.
2. Median from observed: part lead time is offset according to the median lateness in the observed delivery set.

3. Mean from observed: part lead time is offset according to the mean lateness in the observed delivery set.
4. Full buffer from observed: each part has its lead time offset according to the maximum lateness in the observed delivery set.
5. Optimized buffer from observed: the optimized lead time buffer is determined according to a linear program run on the observed delivery set to minimize total cost.
6. Full buffer from fitted distribution: a lognormal distribution is fitted for the lateness of each part. Each part then has its lead time offset by 99.7% of that distribution.
7. Optimized buffer from fitted distribution: 1000 new deliveries are sampled from the fitted lognormal distribution. This new delivery set is used to create an optimized time buffering strategy according to the above linear program.
8. Optimized buffer from created sample: utilizing a data set created by sampling the observed deliveries (sampling the observed deliveries directly as opposed to utilizing the fitted distribution), a third optimized buffering strategy is created with the same linear programming method.

CHAPTER III

COMPUTATIONAL STUDY

In this section, we carry out a comprehensive simulation study to answer the above-listed research questions. For this purpose, we first design the study making sure that all relevant supply chain features are captured to represent the full spectrum of supply chains encountered in practice. This requires the identification of the main parameters that define the supply chain and the ranges within which they vary in practice. Second, we develop a MATLAB program to generate the various supply chain settings, implement the various buffering strategies, and then evaluate and compare their performance. The cost and assembly delivery performance across the multiple settings is then analyzed for both aggregate and individual settings to understand the relative benefit of the various strategies and identify the settings where they are dominant.

3.1 Simulation Design

To evaluate the robustness of the buffering methodology proposed above we created a MATLAB simulation that could simulate the three phases necessary to the business process: observe prior component deliveries, develop buffering strategies based on those observed deliveries, and test those buffering strategies against unobserved component deliveries. The strategies could then be evaluated against inventory holding cost and final assembly delivery performance.

3.1.1 Step 1

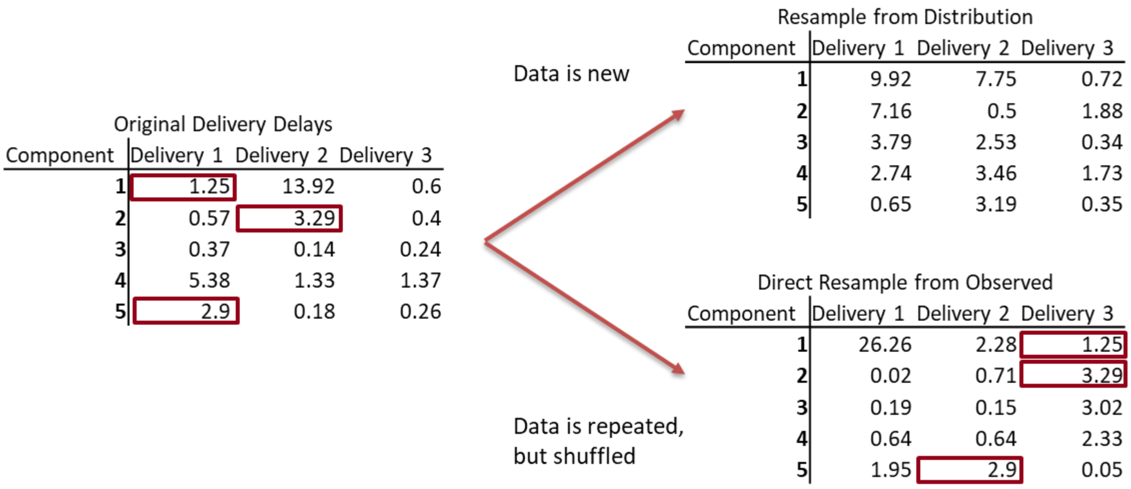
The first thing the simulation must accomplish is to simulate several deliveries of all of the components necessary for the assembly so that their lateness performance can be observed. To do this the simulation first establishes lateness parameters for each individual component according to a lognormal distribution. These lateness parameters are hidden from the code that generates the buffering strategies – those strategies are determined solely from the observed delivery performance that results from this underlying distribution. These lateness parameters are also persistent – a single component maintains its lateness distribution through the entire simulation process, both for the initial observed deliveries and the final deliveries used to evaluate performance.

Once each component has been assigned a delay distribution a series of deliveries is simulated representing the deliveries required to support some number of assemblies. For each delivery a random lateness is generated for each component according to its specific distribution. This is used to feed the generation of buffer strategies.

Additionally, each component is assigned a specific holding cost according to pre-defined parameters. This holding cost is used for certain time buffering strategies and in the final evaluation of performance.

3.1.2 Sampling

Some of the buffering strategies we evaluated are based on only the observed deliveries, while others utilize resampling techniques to hopefully provide better insight from limited data (Chapter 2). Two different resampling techniques were used.



Original Delivery Delays

Component	Delivery 1	Delivery 2	Delivery 3
1	1.25	13.92	0.6
2	0.57	3.29	0.4
3	0.37	0.14	0.24
4	5.38	1.33	1.37
5	2.9	0.18	0.26

Data is new

Resample from Distribution

Component	Delivery 1	Delivery 2	Delivery 3
1	9.92	7.75	0.72
2	7.16	0.5	1.88
3	3.79	2.53	0.34
4	2.74	3.46	1.73
5	0.65	3.19	0.35

Data is repeated, but shuffled

Direct Resample from Observed

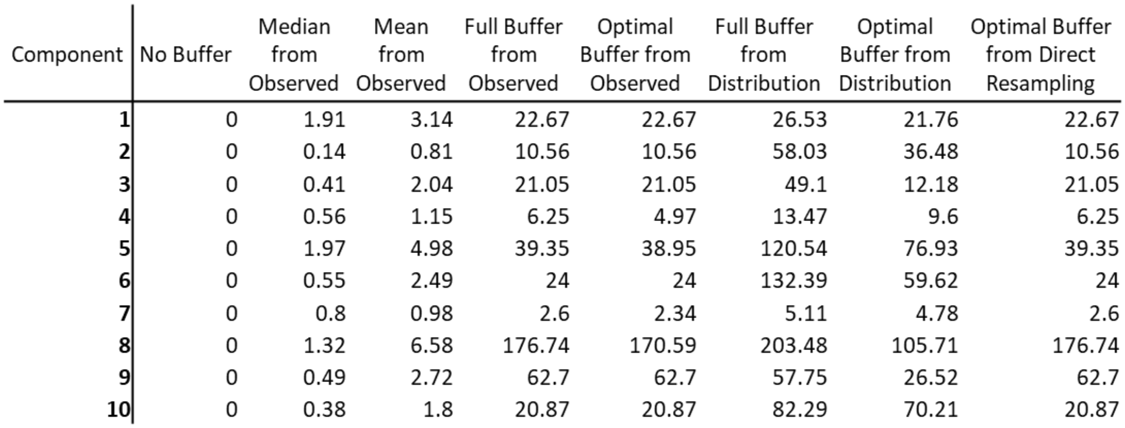
Component	Delivery 1	Delivery 2	Delivery 3
1	26.26	2.28	1.25
2	0.02	0.71	3.29
3	0.19	0.15	3.02
4	0.64	0.64	2.33
5	1.95	2.9	0.05

Figure 6 – An example showing the two different resampling strategies

The first strategy is to first fit a lognormal distribution to the observed lateness of each component in an attempt to determine the underlying lateness distribution. This is shown in the top-right portion of Figure 6. The second strategy is to randomly resample the observed lateness for each delivery for each component, the goal being to preserve the underlying distribution (where a fitted distribution may be incorrect) while still generating new delivery combinations between the components. This second strategy is shown in the bottom-right portion of Figure 6. Note in Figure 6 that in the bottom-right we see repeats of the same data points (boxed in red, the table shown is a subset of a larger table so the paired data for the other indices is not visible here) but these data points are in different combinations between the components, while the top-right portion has entirely new lateness values determined from the fitted distribution. Each sampling technique was used to create 1000 additional samples that were used to determine buffer strategies.

3.1.3 Step 2

Following the creation of the new, sampled, data sets, each component is assigned eight separate time buffers according to the defined strategies.



Component	No Buffer	Median from Observed	Mean from Observed	Full Buffer from Observed	Optimal Buffer from Observed	Full Buffer from Distribution	Optimal Buffer from Distribution	Optimal Buffer from Direct Resampling
1	0	1.91	3.14	22.67	22.67	26.53	21.76	22.67
2	0	0.14	0.81	10.56	10.56	58.03	36.48	10.56
3	0	0.41	2.04	21.05	21.05	49.1	12.18	21.05
4	0	0.56	1.15	6.25	4.97	13.47	9.6	6.25
5	0	1.97	4.98	39.35	38.95	120.54	76.93	39.35
6	0	0.55	2.49	24	24	132.39	59.62	24
7	0	0.8	0.98	2.6	2.34	5.11	4.78	2.6
8	0	1.32	6.58	176.74	170.59	203.48	105.71	176.74
9	0	0.49	2.72	62.7	62.7	57.75	26.52	62.7
10	0	0.38	1.8	20.87	20.87	82.29	70.21	20.87

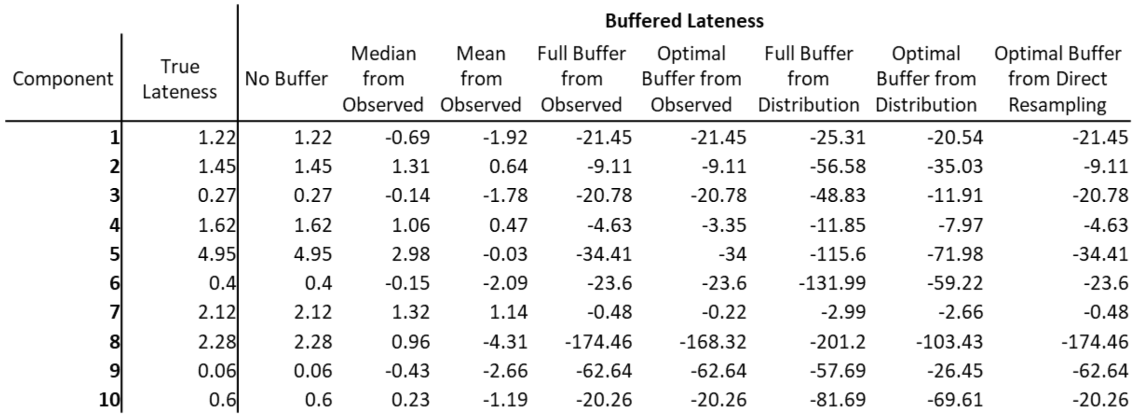
Figure 7 – A sample of the buffer time generated for ten components according to the above strategies.

This time buffer represents the additional ordering time that will be applied to that component (e.g. from Figure 7, the Mean from Observed strategy suggests that Component 5 should be ordered approximately five time units early).

3.1.4 Step 3

Following the assignment of individual buffer times to each component for each strategy we can evaluate the effectiveness of each strategy against our two performance metrics: inventory holding cost and assembly lateness.

To do this, we simulate the lateness of each component for 1000 future deliveries according to each component's individual lateness distribution from Step 1. Once the random lateness is determined for each component for each of the 1000 deliveries, this lateness is offset by each assigned buffer time to determine, for each delivery, what the assembly lateness is (which is equal to the lateness of the latest component) and what the inventory holding cost is for each strategy. This process is demonstrated for a single delivery in Figure 8.



Component	True Lateness	Buffered Lateness No Buffer	Median from Observed	Mean from Observed	Full Buffer from Observed	Optimal Buffer from Observed	Full Buffer from Distribution	Optimal Buffer from Distribution	Optimal Buffer from Direct Resampling
1	1.22	1.22	-0.69	-1.92	-21.45	-21.45	-25.31	-20.54	-21.45
2	1.45	1.45	1.31	0.64	-9.11	-9.11	-56.58	-35.03	-9.11
3	0.27	0.27	-0.14	-1.78	-20.78	-20.78	-48.83	-11.91	-20.78
4	1.62	1.62	1.06	0.47	-4.63	-3.35	-11.85	-7.97	-4.63
5	4.95	4.95	2.98	-0.03	-34.41	-34	-115.6	-71.98	-34.41
6	0.4	0.4	-0.15	-2.09	-23.6	-23.6	-131.99	-59.22	-23.6
7	2.12	2.12	1.32	1.14	-0.48	-0.22	-2.99	-2.66	-0.48
8	2.28	2.28	0.96	-4.31	-174.46	-168.32	-201.2	-103.43	-174.46
9	0.06	0.06	-0.43	-2.66	-62.64	-62.64	-57.69	-26.45	-62.64
10	0.6	0.6	0.23	-1.19	-20.26	-20.26	-81.69	-69.61	-20.26

Figure 8 – Buffered lateness for each component for a single delivery according to the eight buffering strategies

3.2 Simulation Parameters

Several parameters were established for the simulation to evaluate the performance of the eight different strategies in different circumstances:

1. N*: The number of total components required for the assembly.
2. Nvar*: The percentage of the components with a variable lead time. It was found working with the industry partner in the original research that many components arrived consistently on time. We included this parameter to examine the effect of this variation on the effectiveness of the different strategies.
3. Mu*: The mean of the lognormal lead time distribution for a given component.
4. CoV*: The coefficient of variation of the lognormal lead time distribution for a given component.
5. Vhc*: The holding cost of a component with a variable lead time. Holding costs of components that did not have a variable delivery lead time were not of specific interest

and so were fixed to one unit per time unit (the specific units being irrelevant for testing). The vhc parameter allowed us to adjust the relative cost of the variable components (more or less expensive than the non-variable components) in case this had an effect on the effectiveness of different strategies.

6. Obs Del*: The number of historic (observed) deliveries generated in Step 1 of the simulation which would be used to determine the specific buffering times according to the eight strategies.
7. S: The number of times a specific case (a unique set of parameters) is repeated to mitigate outliers in performance caused by randomization. During testing the results were found to be incredibly robust and so a relatively small S of ten was chosen to aid computing speed.
8. New Samples: The number of new "observed deliveries" generated by the sampling in Step 1. This was fixed to 1000.
9. New Deliveries: The number of future deliveries used to evaluate the performance of each buffering strategy in Step 3. This was fixed to 1000.

The parameters marked with an asterisk indicate the parameters of interest. The effects of these parameters on the effectiveness of the different buffering strategies are what we wanted to determine with this research, and so these are the parameters that were varied so that we could study the impact.

3.3 Simulation Strategy

We utilized two separate approaches to running simulations and evaluating the results: 1) a full experimental design to characterize the average performance under various parameter settings, and 2) a detailed study of particular cases of interest that focuses on comparing the full distribution of cost and delivery outcomes associated with the various buffering strategies.

3.3.1 Design of Experiments: Average Performance Analysis

We run a comprehensive experimental design to explore the robustness of the various strategies to provide good average performance across a variety of settings and various parameter values. The results will be analyzed in the aggregate to allow for comparison of the performance of the various strategies as the parameters settings change.

The following parameter variations will be utilized:

1. Number of Components: 10, 100, 500, 1000
2. Percent of Components with Variable Lead Time: 0.1%, 1%, 10%, 25%, 75%, 100%
3. Observed Deliveries: 10, 50, 100
4. Mean of Lateness Distribution for each Component: 0.1-1, 1-10, 0.1-10
5. Coefficient of Variation for Lateness Distribution for each Component: 0.5-1, 1-10, 0.5-10
6. Holding Cost of Variable Components: 0.1-1, 1-10, 0.1-10

Where ranges are listed, these represent allowable upper and lower bounds. Because not all parameters are fixed, we generate 10 replications for each combination of fixed parameters to ensure the results represent a broad set of settings. Within a particular replication, each individual component receives a value for these parameters determined from a uniform random distribution between the listed bounds. Each of the parameters was varied against each other parameter, providing a complete set of 19,440 cases; there are 1,944 parameter combinations and each is replicated 10 times, as mentioned above. Once a case is generated, the component buffers for each of the eight strategies are calculated for those parameters and their performance evaluated.

The first approach was to perform mass aggregate analysis, varying each parameter in relation to all other varied parameters for a total of 19,440 unique parameter sets (cases). This analysis allowed us to understand the impact not just of each parameter but also how different combinations of parameters would impact the effectiveness of each buffering strategy. For example, perhaps a particular strategy was very effective when a large percentage of components was variable, but not if those components had low individual delivery variation. The drawback to the large aggregate analysis is that the sheer volume of data limited the insights we could derive from these simulations to only statistics that could be aggregated, such as mean, median, minimum, and maximum values of our two performance metrics.

3.3.2 Individual Case Analysis: Comparison of Cost and Delivery Distributions

In practice, each individual case could have complex and nuanced behavior which was obscured by the simple aggregate statistics. As a result of this we used the aggregate analysis to guide us to particular cases of interest that could be run individually, allowing us to analyze these cases in detail. Instead of limiting ourselves to, for example, the average holding cost of a buffering strategy across all the deliveries in a particular case, we can display the full distribution of costs given by the 1000 delivery delay scenarios used for evaluation. This provides much greater insight into the specific performance of these cases and allows us to identify which strategies lead to more robust, consistent performance.

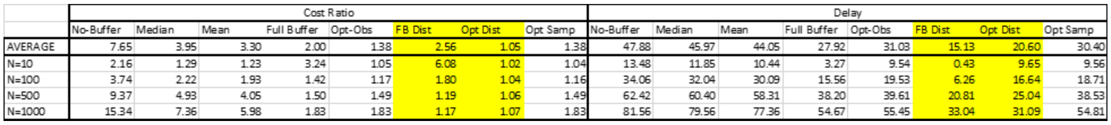
CHAPTER IV

COMPUTATIONAL RESULTS

The full experimental design and the detailed analysis of specific cases described in the previous section will provide deep insight into the performance of the various buffering strategies as the various parameters change and allow us to draw important conclusions. In the next two sections, we report the results and discuss the findings.

4.1 Average Performance Results

The full experiment evaluated a total of 19440 cases, 10 replications of each of the 1944 specific supply chain settings considered. The average performance of the eight strategies is summarized in Figure 9. The left side displays the average "Cost Ratio" and the right side the average "Delay", across all of the cases tested. The "Cost Ratio" for each particular case and strategy is calculated as the average cost incurred when following that strategy in the 1000 scenarios used to evaluate performance, over the best average cost achieved by any of the strategies for that particular case. The "Delay" for each particular case and strategy is the average assembly delivery delay observed in the 1000 scenarios when following that strategy.



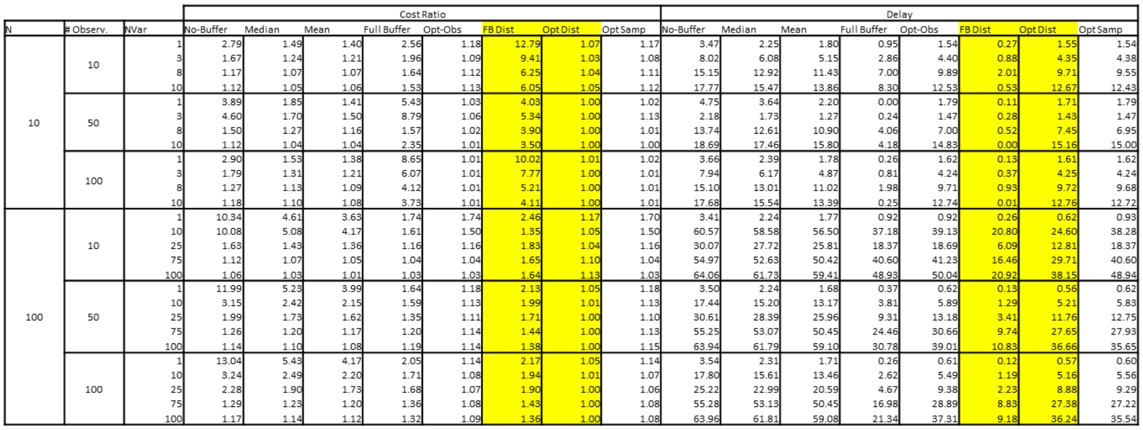
	Cost Ratio								Delay							
	No-Buffer	Median	Mean	Full Buffer	Opt-Obs	FB Dist	Opt Dist	Opt Samp	No-Buffer	Median	Mean	Full Buffer	Opt-Obs	FB Dist	Opt Dist	Opt Samp
AVERAGE	7.65	3.95	3.30	2.00	1.38	2.56	1.05	1.38	47.88	45.97	44.05	27.92	31.03	15.13	20.60	30.40
N=10	2.16	1.29	1.23	3.24	1.05	6.08	1.02	1.04	13.48	11.85	10.44	3.27	9.54	0.43	9.65	9.56
N=100	3.74	2.22	1.93	1.42	1.17	1.80	1.04	1.16	34.06	32.04	30.09	15.56	19.53	6.26	16.64	18.71
N=500	9.37	4.93	4.05	1.50	1.49	1.19	1.06	1.49	62.42	60.40	58.31	38.20	39.61	20.81	25.04	38.53
N=1000	15.34	7.36	5.98	1.83	1.83	1.17	1.07	1.83	81.56	79.56	77.36	54.67	55.45	33.04	31.09	54.81

Figure 9 – Aggregate cost and delay performance across the 19440 cases tested. The costs listed are a normalized ratio between strategies, where a higher number indicates higher costs in relation to the other strategies. The delay values are absolute values.

Observe that:

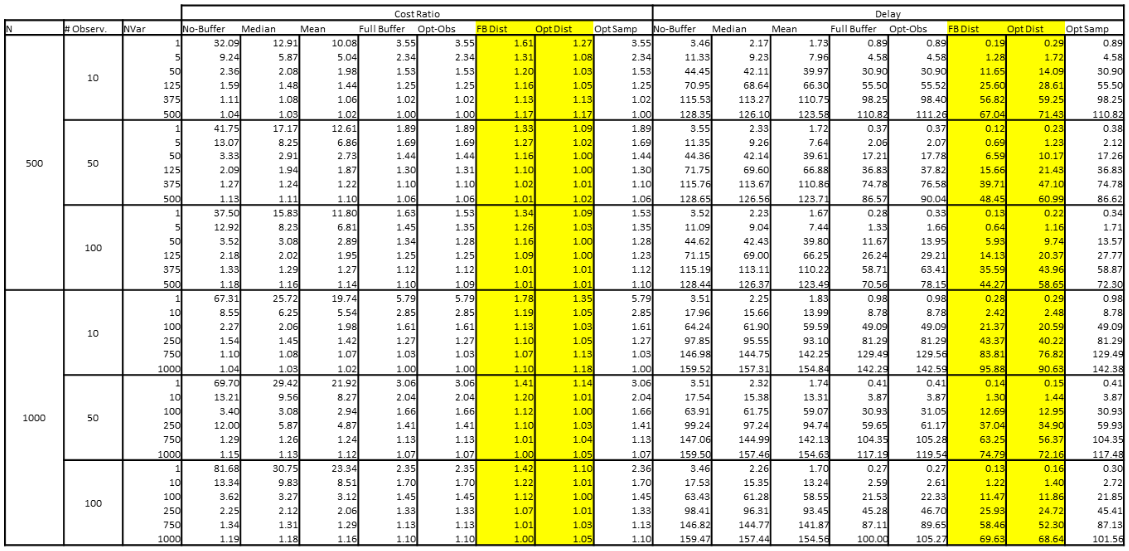
- The strategy that optimizes the buffers assuming the fitted lognormal distribution consistently provides the best cost performance and high delivery performance
- Full buffer (to 3σ) strategies based on the fitted lognormal distribution consistently provide the best delivery performance for moderate number of parts in the assembly. As we shall see, for N≥ 1000 more aggressive buffering may be needed and the optimized strategy performs better. The superior delivery performance, however, comes at a high cost for for assemblies with low number of components. As the number of components rises, the cost becomes quite similar to that of the optimized strategy. Full buffering strategies are thus found to be very competitive for high numbers of components.
- As the number of components rises, using the fitted distribution rather than the observed samples to make buffering decisions is a must. Both the cost and delay performance of the resulting strategies (FB Dist and Opt Dist) become increasingly superior to their counterparts (Full Buffer and Opt Obs or Opt Samp).
- Surprisingly, the performance of the OptSamp strategy, which finds the optimized buffers using 1000 scenarios generated from independently resampling from each individual part observed deliveries, is virtually identical to that of the OptObs strategy, which optimizes only using the few observed scenarios, a much-reduced set of cases. There is only a slight improvement in average delivery delay gained by the resampling.

Let's explore further how these general tendencies are impacted by each of the important parameters that have been varied in the experiment. Figure 9 shows that *N*, the number of components, has a large impact on relative performance of the various strategies. Two other main parameters of interest are a) *# of Obs*. the number of observed deliveries that the full buffer and optimized strategies are based on, and b) *NVar*, the number of components with unreliable deliveries out of the entire set of components in the assembly. Figures 10 and 11 report the performance, Cost Ratio and Delay, as the three parameters (*N*, *# of Obs, and NVar*) vary.



			Cost Ratio								Delay							
N	# Observ.	NVar	No-Buffer	Median	Mean	Full Buffer	Opt-Obs	FB Dist	Opt Dist	Opt Samp	No-Buffer	Median	Mean	Full Buffer	Opt-Obs	FB Dist	Opt Dist	Opt Samp
10	10	1	2.79	1.49	1.40	2.56	1.18	12.79	1.07	1.17	3.47	2.25	1.80	0.95	1.54	0.27	1.55	1.54
		3	1.67	1.24	1.21	1.96	1.09	9.41	1.03	1.08	8.02	6.08	5.15	2.86	4.40	0.88	4.35	4.38
		8	1.17	1.07	1.07	1.64	1.12	6.25	1.04	1.11	15.15	12.92	11.43	7.00	9.89	2.01	9.71	9.55
		10	1.12	1.05	1.06	1.53	1.13	6.05	1.05	1.12	17.77	15.47	13.86	8.30	12.53	0.53	12.67	12.43
	50	1	3.89	1.85	1.41	5.43	1.03	4.03	1.00	1.02	4.75	3.64	2.20	0.00	1.79	0.11	1.71	1.79
		3	4.60	1.70	1.50	8.79	1.06	5.34	1.00	1.13	2.18	1.73	1.27	0.24	1.47	0.28	1.43	1.47
		8	1.50	1.27	1.16	1.57	1.02	3.90	1.00	1.01	13.74	12.61	10.90	4.06	7.00	0.52	7.45	6.95
		10	1.12	1.04	1.04	2.35	1.01	3.50	1.00	1.00	18.69	17.46	15.80	4.18	14.83	0.00	15.16	15.00
	100	1	2.90	1.53	1.38	8.65	1.01	10.02	1.01	1.02	3.66	2.39	1.78	0.26	1.62	0.13	1.61	1.62
		3	1.79	1.31	1.21	6.07	1.01	7.77	1.00	1.01	7.94	6.17	4.87	0.81	4.24	0.37	4.25	4.24
		8	1.27	1.13	1.09	4.12	1.01	5.21	1.00	1.01	15.10	13.01	11.02	1.98	9.71	0.93	9.72	9.68
		10	1.18	1.10	1.08	3.73	1.01	4.11	1.00	1.01	17.68	15.54	13.39	0.25	12.74	0.01	12.76	12.72
100	10	1	10.34	4.61	3.63	1.74	1.74	2.46	1.17	1.70	3.41	2.24	1.77	0.92	0.92	0.26	0.62	0.93
		10	10.08	5.08	4.17	1.61	1.50	1.35	1.05	1.50	60.57	58.58	56.50	37.18	39.13	20.80	24.60	38.28
		25	1.63	1.43	1.36	1.16	1.16	1.83	1.04	1.16	30.07	27.72	25.81	18.37	18.69	6.09	12.81	18.37
		75	1.12	1.07	1.05	1.04	1.04	1.65	1.10	1.04	54.97	52.63	50.42	40.60	41.23	16.46	29.71	40.60
		100	1.06	1.03	1.01	1.03	1.03	1.64	1.13	1.03	64.06	61.73	59.41	48.93	50.04	20.92	38.15	48.94
	50	1	11.99	5.23	3.99	1.64	1.18	2.13	1.05	1.18	3.50	2.24	1.68	0.37	0.62	0.13	0.56	0.62
		10	3.15	2.42	2.15	1.59	1.13	1.99	1.01	1.13	17.44	15.20	13.17	3.81	5.89	1.29	5.21	5.83
		25	1.99	1.73	1.62	1.35	1.11	1.71	1.00	1.10	30.61	28.39	25.96	9.31	13.18	3.41	11.76	12.75
		75	1.26	1.20	1.17	1.20	1.14	1.44	1.00	1.13	55.25	53.07	50.45	24.46	30.66	9.74	27.65	27.93
		100	1.14	1.10	1.08	1.19	1.14	1.38	1.00	1.15	63.94	61.79	59.10	30.78	39.01	10.83	36.66	35.65
	100	1	13.04	5.43	4.17	2.05	1.14	2.17	1.05	1.14	3.54	2.31	1.71	0.26	0.61	0.12	0.57	0.60
		10	3.24	2.49	2.20	1.71	1.08	1.94	1.01	1.07	17.80	15.61	13.46	2.62	5.49	1.19	5.16	5.56
		25	2.28	1.90	1.73	1.68	1.07	1.90	1.00	1.06	25.22	22.99	20.59	4.67	9.38	2.23	8.88	9.29
		75	1.29	1.23	1.20	1.36	1.08	1.43	1.00	1.08	55.28	53.13	50.45	16.98	28.89	8.83	27.38	27.22
		100	1.17	1.14	1.12	1.32	1.09	1.36	1.00	1.08	63.96	61.81	59.08	21.34	37.31	9.18	36.24	35.54

Figure 10 – Aggregate simulation results for cases where N = 10 and N = 100. The costs listed are a normalized ratio between strategies, where a higher number indicates higher costs in relation to the other strategies. The delay values are absolute values. All values are based on the mean performance of each strategy for all iterations of each particular parameter set.



			Cost Ratio								Delay							
N	# Observ.	NVar	No-Buffer	Median	Mean	Full Buffer	Opt-Obs	FB Dist	Opt Dist	Opt Samp	No-Buffer	Median	Mean	Full Buffer	Opt-Obs	FB Dist	Opt Dist	Opt Samp
500	10	1	32.09	12.91	10.08	3.55	3.55	1.61	1.27	3.55	3.46	2.17	1.73	0.89	0.89	0.19	0.29	0.89
		5	9.24	5.87	5.04	2.34	2.34	1.31	1.08	2.34	11.33	9.23	7.96	4.58	4.58	1.28	1.72	4.58
		50	2.36	2.08	1.98	1.53	1.53	1.20	1.03	1.53	44.45	42.11	39.97	30.90	30.90	11.65	14.09	30.90
		125	1.59	1.48	1.44	1.25	1.25	1.16	1.05	1.25	70.95	68.64	66.30	55.50	55.52	25.60	28.61	55.50
		375	1.11	1.08	1.06	1.02	1.02	1.13	1.13	1.02	115.53	113.27	110.75	98.25	98.40	56.82	59.25	98.25
		500	1.04	1.03	1.02	1.00	1.00	1.17	1.17	1.00	128.35	126.10	123.58	110.82	111.26	67.04	71.43	110.82
	50	1	41.75	17.17	12.61	1.89	1.89	1.33	1.09	1.89	3.55	2.33	1.72	0.37	0.37	0.12	0.23	0.38
		5	13.07	8.25	6.86	1.69	1.69	1.27	1.02	1.69	11.35	9.26	7.64	2.06	2.07	0.69	1.23	2.12
		50	3.33	2.91	2.73	1.44	1.44	1.16	1.00	1.44	44.36	42.14	39.61	17.21	17.78	6.59	10.17	17.26
		125	2.09	1.94	1.87	1.30	1.31	1.10	1.00	1.30	71.75	69.60	66.88	36.83	37.82	15.66	21.43	36.83
		375	1.27	1.24	1.22	1.10	1.10	1.02	1.01	1.10	115.76	113.67	110.86	74.78	76.58	39.71	47.10	74.78
		500	1.13	1.11	1.10	1.06	1.06	1.01	1.02	1.06	128.65	126.56	123.71	86.57	90.04	48.45	60.99	86.62
	100	1	37.50	15.83	11.80	1.63	1.53	1.34	1.09	1.53	3.52	2.23	1.67	0.28	0.33	0.13	0.22	0.34
		5	12.92	8.23	6.81	1.45	1.35	1.26	1.03	1.35	11.09	9.04	7.44	1.33	1.66	0.64	1.16	1.71
		50	3.52	3.08	2.89	1.34	1.28	1.16	1.00	1.28	44.62	42.43	39.80	11.67	13.95	5.93	9.74	13.57
		125	2.18	2.02	1.95	1.25	1.25	1.09	1.00	1.23	71.15	69.00	66.25	26.24	29.21	14.13	20.37	27.77
		375	1.33	1.29	1.27	1.12	1.12	1.01	1.01	1.12	115.19	113.11	110.22	58.71	63.41	35.59	43.96	58.87
		500	1.18	1.16	1.14	1.10	1.09	1.01	1.01	1.10	128.44	126.37	123.49	70.56	78.15	44.27	58.65	72.30
1000	10	1	67.31	25.72	19.74	5.79	5.79	1.78	1.35	5.79	3.51	2.25	1.83	0.98	0.98	0.28	0.29	0.98
		10	8.55	6.25	5.54	2.85	2.85	1.19	1.05	2.85	17.96	15.66	13.99	8.78	8.78	2.42	2.48	8.78
		100	2.27	2.06	1.98	1.61	1.61	1.13	1.03	1.61	64.24	61.90	59.59	49.09	49.09	21.37	20.59	49.09
		250	1.54	1.45	1.42	1.27	1.27	1.10	1.05	1.27	97.85	95.55	93.10	81.29	81.29	43.37	40.22	81.29
		750	1.10	1.08	1.07	1.03	1.03	1.07	1.13	1.03	146.98	144.75	142.25	129.49	129.56	83.81	76.82	129.49
		1000	1.04	1.03	1.02	1.00	1.00	1.10	1.18	1.00	159.52	157.31	154.84	142.29	142.59	95.88	90.63	142.38
	50	1	69.70	29.42	21.92	3.06	3.06	1.41	1.14	3.06	3.51	2.32	1.74	0.41	0.41	0.14	0.15	0.41
		10	13.21	9.56	8.27	2.04	2.04	1.20	1.01	2.04	17.54	15.38	13.31	3.87	3.87	1.30	1.44	3.87
		100	3.40	3.08	2.94	1.66	1.66	1.12	1.00	1.66	63.91	61.75	59.07	30.93	31.05	12.69	12.95	30.93
		250	12.00	5.87	4.87	1.41	1.41	1.10	1.03	1.41	99.24	97.24	94.74	59.65	61.17	37.04	34.90	59.93
		750	1.29	1.26	1.24	1.13	1.13	1.01	1.04	1.13	147.06	144.99	142.13	104.35	105.28	63.25	56.37	104.35
		1000	1.15	1.13	1.12	1.07	1.07	1.00	1.05	1.07	159.50	157.46	154.63	117.19	119.54	74.79	72.16	117.48
	100	1	81.68	30.75	23.34	2.35	2.35	1.42	1.10	2.36	3.46	2.26	1.70	0.27	0.27	0.13	0.16	0.30
		10	13.34	9.83	8.51	1.70	1.70	1.22	1.01	1.70	17.53	15.35	13.24	2.59	2.61	1.22	1.40	2.72
		100	3.62	3.27	3.12	1.45	1.45	1.12	1.00	1.45	63.43	61.28	58.55	21.53	22.33	11.47	11.86	21.85
		250	2.25	2.12	2.06	1.33	1.33	1.07	1.01	1.33	98.41	96.31	93.45	45.28	46.70	25.93	24.72	45.41
		750	1.34	1.31	1.29	1.13	1.13	1.01	1.03	1.13	146.82	144.77	141.87	87.11	89.65	58.46	52.30	87.13
		1000	1.19	1.18	1.16	1.10	1.10	1.00	1.05	1.10	159.47	157.44	154.56	100.00	105.27	69.63	68.64	101.56

Figure 11 – Aggregate simulation results for cases where N = 500 and N = 1000. The values listed are derived in a similar fashion to those in Figure 8.

In the low-N cases in Figure 10 the aggressive buffering strategies (the two Full Buffer strategies) called for in the prior research perform very poorly compared to the other strategies. While the assembly delay performance is expectedly good, the costs of the two Full Buffer strategies are very poor, even when compared to the No Buffer case.

Recall the two key points from Chapter 1: 1) Reducing the lateness of the latest component has a disproportionate effect on the total holding cost compared to reducing the earliness of the earliest part; when the number of components is low, however, the costs of holding one part when early vs. holding the rest of the assembly when being last are not as different. 2) As the number of components increases so does the likelihood that any one component will arrive as an outlier on its delivery distribution. Effectively, when there are a large number of components at least one is very likely to be an outlier, and that outlier will incur a huge cost. An aggressive buffering strategy is warranted as it is effectively betting that there will be an outlier and mitigating appropriately. If no outlier occurs, then an aggressive buffering strategy will result in over-buffering, and the inventory cost will be driven by the buffering rather than the lateness. Consider N = 10 as shown in Figure 10. With only ten components, it is exceedingly unlikely that a severe outlier will occur on any given component, even if all ten components are allowed to vary (though you will note the relative cost performance of aggressive buffering strategies improves as Nvar increases). As such, an aggressive buffering strategy should result in a cost increase, as is shown here. Likewise, as the number of observed deliveries increases, it is more likely to observe later and later components, causing the Full Buffer from Observed Data strategy to become increasingly more expensive. Conversely, the No Buffer strategy assumes that components will be on-time. While this is clearly a poor assumption in any case with observed delivery variation, in the cases with a low-N this is still a better assumption than that underlying an aggressive buffer. With very few components, it is more likely that all components will be closer to on-time than that any one component will be exceedingly late; in the rare case of a very late component, the inventory cost associated with carrying the few other components as they wait for the pacing part is not prohibitive compared with the extra cost of aggressively buffering that component in all other cases.

Contrast this with the cost performance seen in Figure 11. As N increases, so too does the performance of our aggressive buffering strategies. There are two important things to note; not only do the Full Buffer strategies begin to consistently improve, but their corresponding Optimized strategies (obtained by solving the stochastic program on the same data set that the two Full Buffer strategies utilized) begin to perform more similarly to the Full Buffer strategies. This is reflected in many cases with a high N where the Optimized strategy's performance was in fact the same as the Full Buffer.

One other point of interest is the Opt Samp column, representing the performance of the strategy Optimized from the resampling of the observed data (as opposed to the Optimized based on the fitted distribution). Surprisingly, the Optimized from the resampled data performed very similarly to the Optimized from the observed data, even with a low number of observations. The additional scenarios generated from independently sampling from the observed data for each part were expected to lead to a more robust buffering strategy that would perform better when evaluated over the new set of 1000 scenarios. However, they did not fundamentally change the buffering strategy and resulted in almost identical performance. The detailed analysis in the next section further supports this conclusion: The resampling strategy from observed data is ineffective and unnecessary; comparable performance is achieved from direct consideration of the observed scenarios.

In the next section, we move beyond average performance and explore individual cases to gain a better understanding of the range of cost and delivery delay outcomes resulting from each of the buffering strategies in the various supply chain settings.

4.2 Comparison of Cost and Delay Distributions

With the results from the experimental design, we honed in on a few test cases and varying parameter settings to examine more closely. For these cases we generated box plots from the cost and delivery delay incurred for each of the 1000 deliveries in the evaluation set. This allowed us to examine the specific behavior of each strategy in particular cases of interest rather than relying solely on aggregate statistics. How robust are the buffers proposed to provide good performance across the spectrum of potential component delivery scenarios (captured by the 1000 scenarios used for evaluation)?

From the aggregate results we identified four particular cases that warranted the more detailed analysis, as always looking at the impact of both inventory holding cost and final delivery performance:

1. *The effect of the total number of components in the assembly*: From the initial analysis in Chapter 1 it is clear that the more aggressive buffering strategies are warranted where there are more components. We ran a series of detailed tests to examine more closely what the specific behavior is of the different strategies as

the number of components increases, at what point does more aggressive buffering become warranted, and how aggressive should that buffering be.

2. *The effect of the percentage of variable components in the assembly*: The aggressive buffering strategy is justified as late components require holding all other components for increasing amounts of time, disproportionately increasing holding cost while also contributing to assembly lateness. How will this be impacted as the overall variability of the assembly increases? If the majority of components, or even all components, have variable delivery lead times, would aggressive buffering become counterproductive by increasing holding cost as a result of ordering the majority of components very early?
3. *The effect of the variation of late components*: The initial research was performed for a case with high variability of component lead time. How does this variability affect the performance of the different strategies? What happens if component variability is increased or decreased?
4. *The effect of relative holding cost of the variable components vs. reliable components*: Intuitively, the lower the holding cost of the variable components, the more affordable and attractive aggressive buffering of these components is. Will this effect be sufficient to make full buffers cost competitive in cases with low number of components? The aggregate results show that the more aggressive buffering strategies are less effective for a low number of components. This is expected behavior; the aggressive buffering strategies are reliant on the cost disparity between the limited holding cost of a single component being early compared to the higher holding cost of having to hold all components for a late

assembly. In cases where there are a low number of components, this disparity is reduced, and early deliveries have a correspondingly higher effect on the total holding cost. Additionally, smaller number of components reduce the likelihood that there will be an outlier in delivery performance. This creates a situation where parts are consistently arriving early, adding more cost to the total than is offset by the rare cases where the assembly might otherwise be very late. However, if the late components were relatively inexpensive compared to the rest of the assembly, the additional cost incurred by their early arrival will be lessened and would eventually result in more aggressive buffering once again being an effective strategy.

In what follows we explore each of these cases in detail for representative settings.

Effect of the Number of Components on the Cost and Delivery of each Buffering Strategy

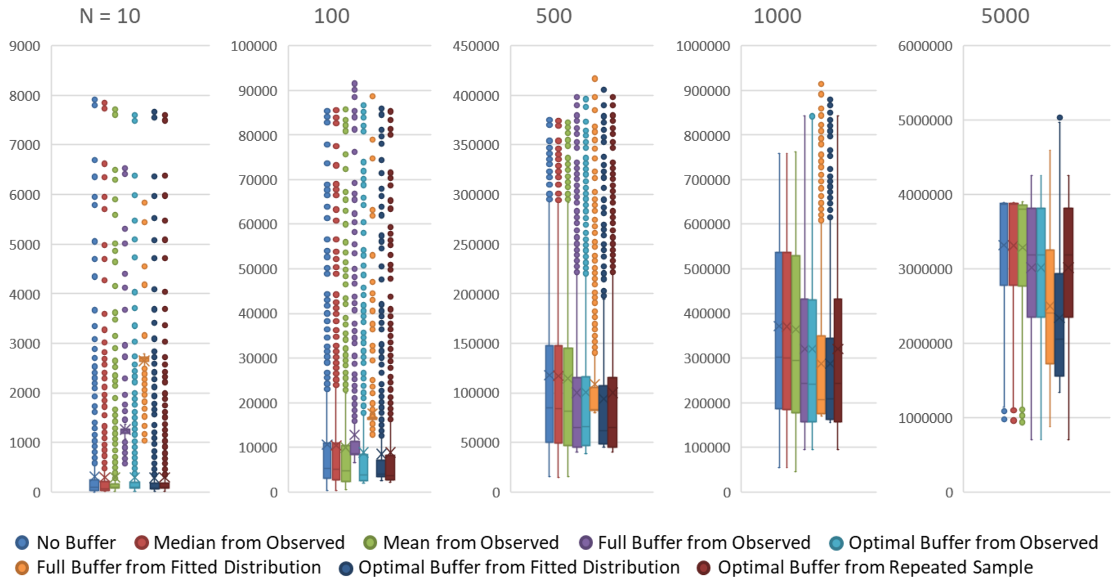


Figure 12 – Effect of the number of components on the cost of each buffering strategy. Parameters: Nvar = 25%, Mu = 0.1-10, CoV = 0.5-10, Vhc = 1-10, 50 Observed Deliveries

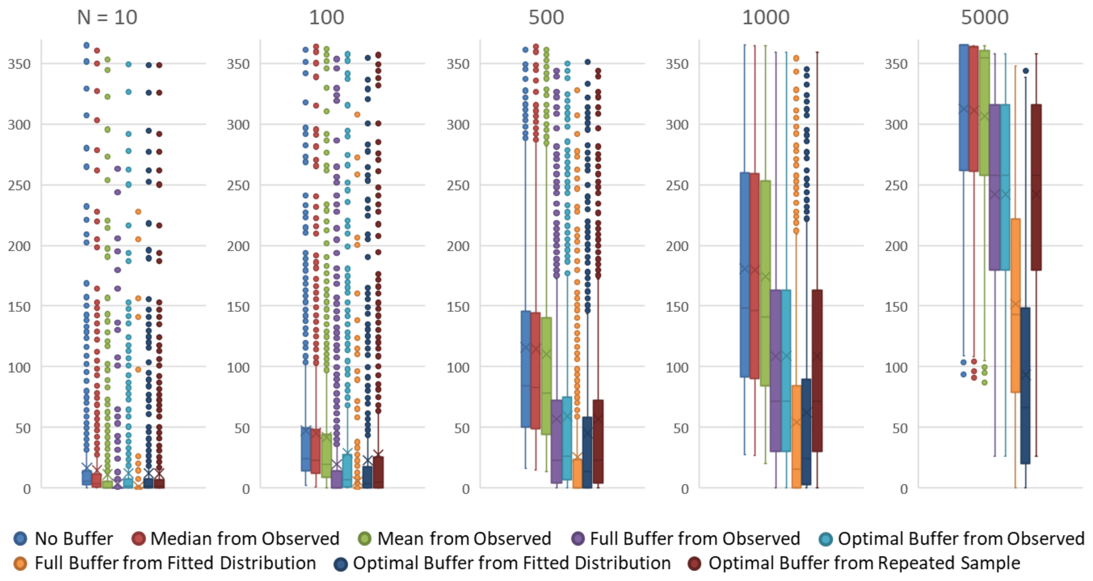


Figure 13 – Effect of the number of components on the delay of the final assembly for each buffering strategy. Parameters as in Figure 11

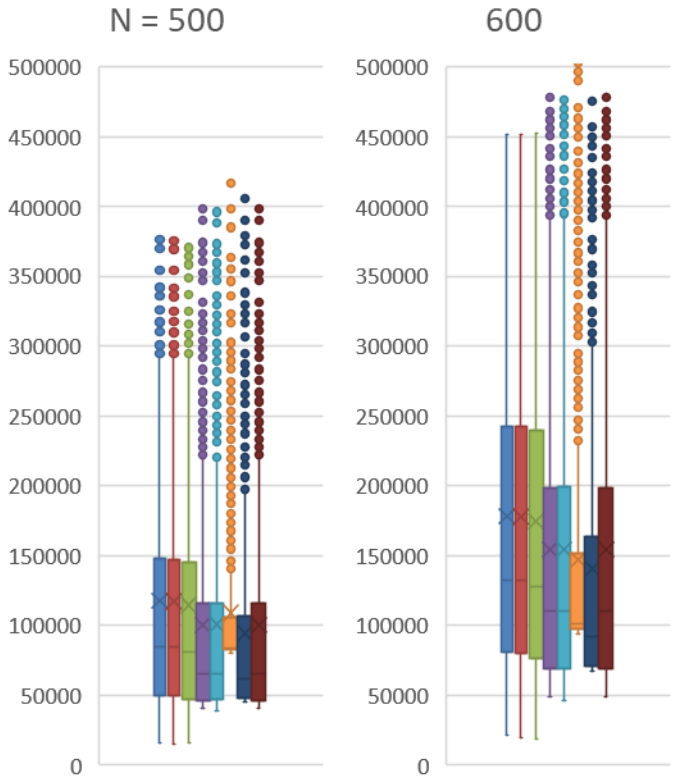


Figure 14 – Effect of the number of components on the cost of each buffering strategy for N = 500 and N = 600. Colors and parameters as in Figures 11 and 12

Figures 12 and 13 show the effect of increasing the number of components on both the cost and the final assembly delay for each strategy. Reinforcing the insights from the aggregate simulations, we see that though the delivery performance of the Full Buffer from the Fitted Distribution is consistently superior until N = 5000, the cost is prohibitively high until past N = 500 (the actual tipping point appears to be around N = 600, see Figure 14). Additionally, we see that the cost and delivery performance for the Full Buffer from the Fitted Distribution begins to lose ground against the Optimized from the Fitted Distribution. Recall that the Full Buffer from the Fitted Distribution is defined as being 3σ of that distribution. It stands to reason from the above that as the number of components increases, so must the aggressiveness of the buffering strategy. This is what we are seeing here; past N = 1000, 3σ ceases to be as effective, and more aggressive buffering is required. Additionally, note the clipping on the top of the first three boxes for N = 5000 in Figure 12. This behavior will be seen in the following figures as well, and is explained in Appendix A.

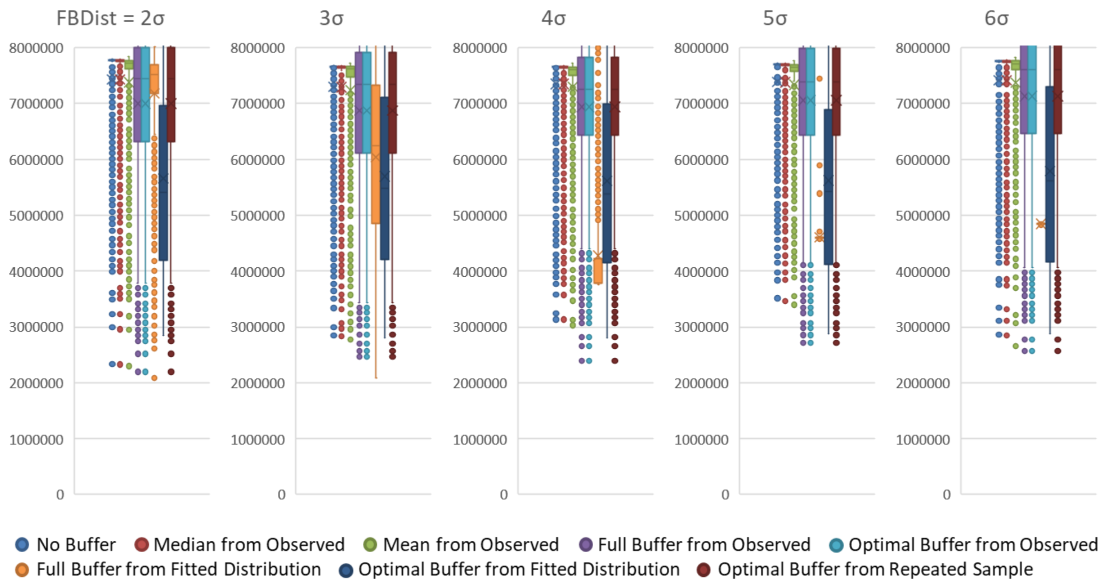


Figure 15 – Effect of increasing the aggressiveness of the Full Buffer from the Fitted Distribution Strategy on the cost of a case with a high number of components. Parameters: N = 10,000, Nvar = 25%, Mu = 0.1-10, Cov = 0.5-10, Vhc = 1-10, 50 Observed Deliveries

As noted in Figure 12, as the number of components increases the 3σ buffer is no longer effective enough to mitigate the increased chance of wild outliers in delivery performance. As N increases, so too must the buffer aggressiveness to keep pace with the increase in compounding probabilities, as discussed in Chapter 1. Figure 15 shows the impact of the aggressiveness of the Full Buffer from the Fitted Distribution Strategy on the inventory holding cost in a case with a very high number of components (N = 10,000). Where the 3σ buffer was effective at and below N = 1000, it was less effective at N = 5000 and even less effective as N further increases. A more aggressive 4σ buffer is, however, very effective, providing both the best cost performance of these strategies and a reduced variability of cost, representing less risk to the business. More aggressive buffering is ineffective in this case; the number of components would likely need to be much higher before a 5σ or 6σ buffering strategy is necessary. In addition, the incredibly long delivery delays suggested by the long tail of the lognormal distribution are unrealistic in practice, since other business processes will be in place to ensure business continuity (alternative supplier, additional capacity, etc).

Effect of the Percentage of Variable Components on the Cost and Delivery of Each Buffering Strategy

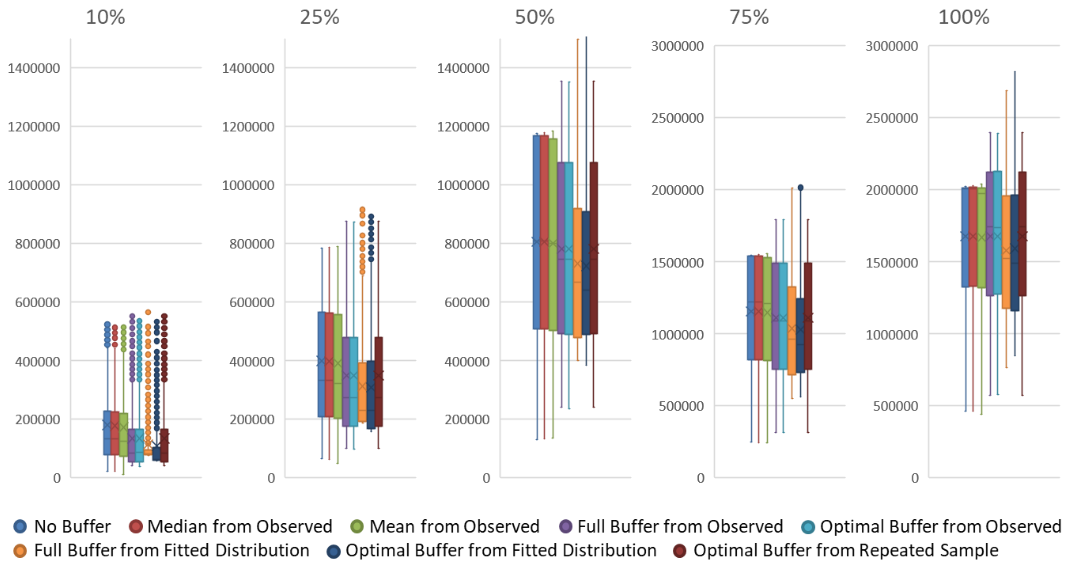


Figure 16 – Effect of the percentage of variable components on the cost of each buffering strategy. Parameters: N = 1000, Mu = 0.1-10, CoV = 0.5-10, Vhc = 1-10, 50 Observed Deliveries

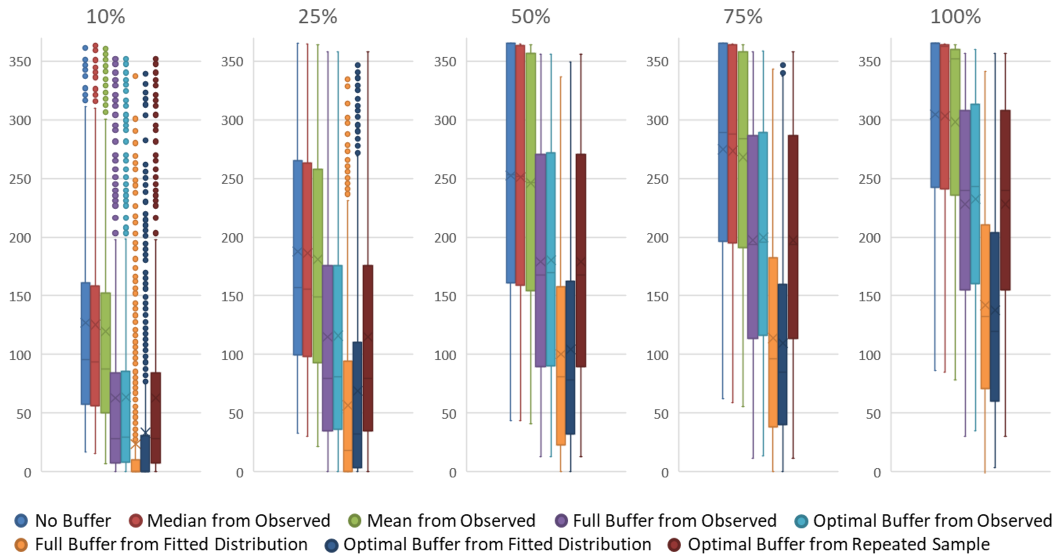


Figure 17 – Effect of the percentage of variable components on the delay of the final assembly for each buffering strategy. Parameters as in Figure 15

Figures 16 and 17 show the effect of increasing the percentage of components in the assembly that have a variable delivery time. As the percentage of variable components increases, we can see that cost increases and delivery performance decreases. However, the relative performance of each strategy remains relatively constant. A strategy that provides superior performance when 10% of the components have variable lead time provides superior performance when 100% of the components have variable lead times.

Effect of the Variation of Late Components on the Cost and Delivery Performance of Each Buffering Strategy

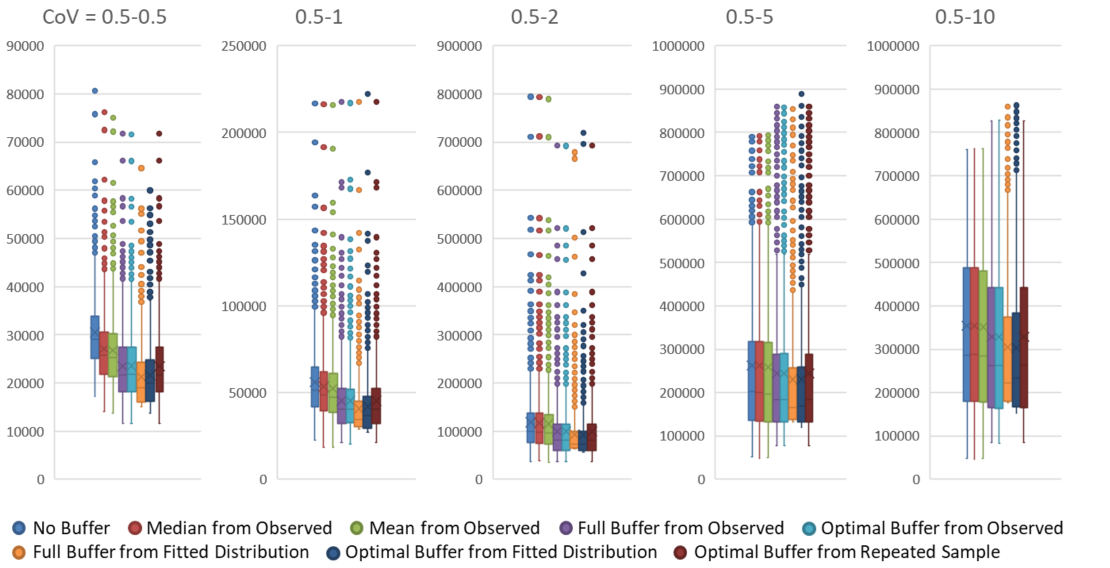


Figure 18 – The effect of the variation of late components on the cost of each buffering strategy. Parameters: N = 1000, Nvar = 25%, Mu = 5, Vhc = 1-10, 50 Observed Deliveries

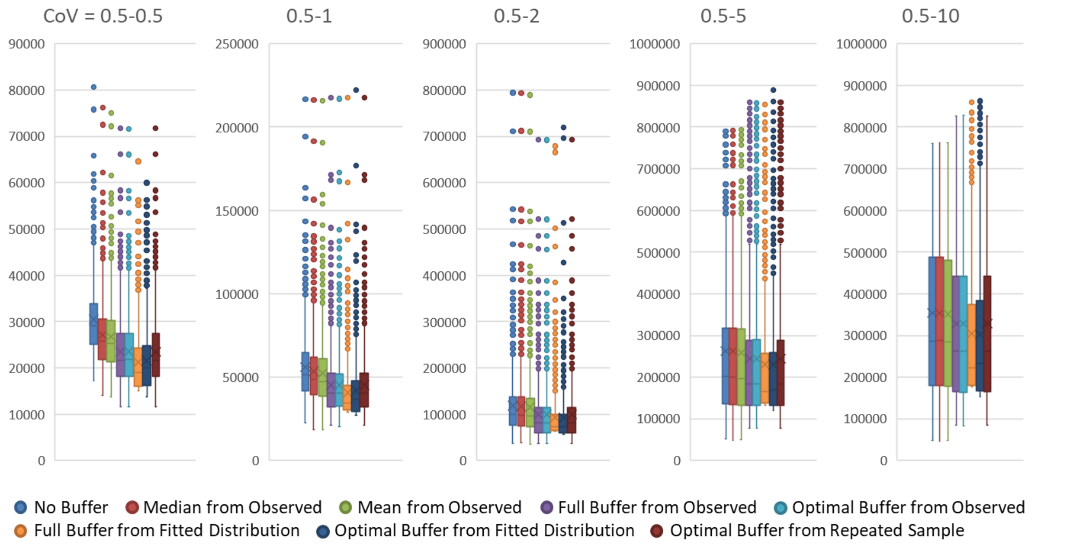


Figure 19 – The effect of the variation of late components on the delay of the final assembly for each buffering strategy. Parameters as in Figure 17

In Figures 18 and 19 we see the impact of increase the variability of late components on both cost and delivery performance for each buffering strategy. Similarly to the impact of increasing the percentage of variable components, here we see that increased variability degrades the effectiveness of all strategies while maintaining their relative position. Once again, increasing the variability does not seem to impact which strategy is most effective.

CHAPTER V

CONCLUSION

The above data supports the central argument of the prior work: for assemblies with a high number of components, an aggressive time buffering strategy both reduces cost and improves delivery performance. Importantly, this remains the case in situations of both high and low variability. We have shown that an aggressive buffering strategy has a robust performance as other parameters vary. Further, as the number of components increases so too must the aggressiveness of the buffering strategy. Where a 3σ buffer was effective at N = 1000, it had already begun to lose ground relative to the optimized strategies at N = 5000; a 4σ buffer becomes more attractive at that point. On the other hand, when N is low (e.g. N=10 or 100), significant time buffering is not cost effective unless the unreliable components that need buffering are very inexpensive. This is because the probability of one of the few components being very late is low and the cost of carrying all other components until the pacing one arrives is relatively low. While there is an impact from other parameters, their effect is more nuanced. Additionally, when choosing a strategy, it is worth pursuing a buffering strategy based on a fitted distribution rather than just on observed data as this was shown to have a consistently superior performance, regardless of preference for an optimized model or a simple percentile model. The optimized model based on the fitted distribution provides the best performance consistently across all cases, but as the number of components increases, the performance of a full buffer strategy becomes very competitive and in many cases more robust (less risky as the range of outcomes is smaller). The more the observations used to fit the distribution, the greater the advantage of these models as the fitted distribution more accurately captures reality.

CHAPTER VI

FURTHER WORK

While this work has shown the robustness and effectiveness of an aggressive buffering strategy, there remains a question of just how aggressive the buffering should be in relation to the number of components in the assembly.

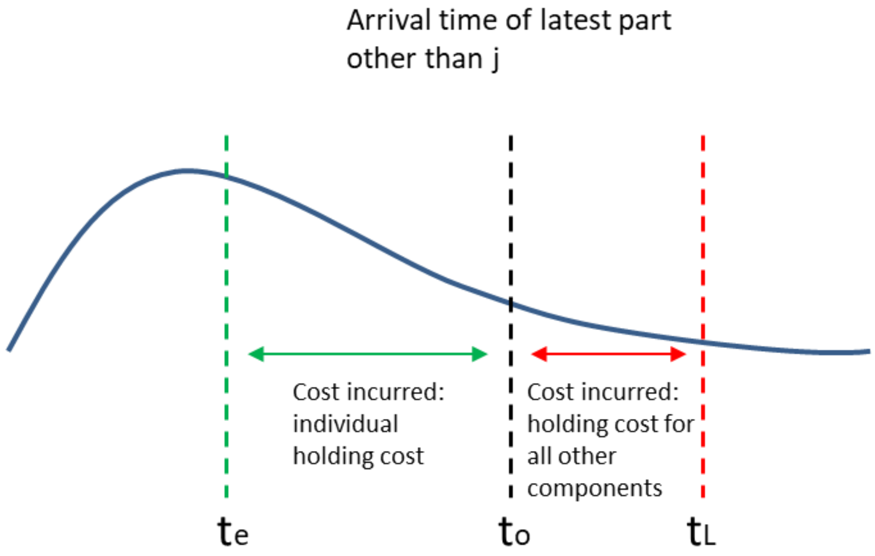

Arrival time of latest part other than j
Cost incurred: individual holding cost
Cost incurred: holding cost for all other components
te
to
tL

Figure 20 – Delivery lateness distribution and cost incurred for a single component

Figure 20 shows a proposed mathematical approach to the optimal buffering problem. Consider a particular component and let t_o be the lateness of the latest component other than the component in question. The initial component may arrive earlier, at a time $t_e < t_o$, or later at a time $t_L > t_{o.}$ In the first case, the early component's holding cost contribution to the overall total is equal to the holding cost of the component times (t_o-t_e). In contrast, in the latter case, the holding cost contribution of the late component is equal to (t_L-t_o) times *the sum of all holding costs for all other components* (recall the discussion around Figure 4 in Chapter 1). It stands to reason that the ideal buffering strategy might be one where the expected value of the marginal costs of being early or being late are balanced. The challenge to this approach is that the precise value of t_o is dependent on the individual lateness variation of every other component; as the number of components increases and their individual delivery variation increases, t_o will move to the right.

BIBLIOGRAPHY

Beladi, F. (2014), *Delivery Performance Assessment Tool for Complex Assemblies.* MS Thesis, University of Massachusetts Amherst, February 2014.

Prokle, M. (2017), *Theory and Practice of Supply Chain Synchronization*. PhD Thesis, University of Massachusetts Amherst, May 2017

Prokle, M., F. Beladi, A. Muriel, R. Subbu, C. Greene, J. Heslin (2018) *Supply Chain Synchronization for Assembled Systems with Long and Variable Component Lead Times*. Working paper, University of Massachusetts Amherst.

APPENDIX A: THE IMPACT OF CLIPPED DISTRIBUTIONS

With the lognormal distributions which underlie all the delivery variation used in the simulation there is a very small chance of having extremely late deliveries. As the fundamental case being argued here is centered around the possibility of extreme outliers, so too can we expect severe, unrealistic outliers during simulation (in testing, components were observed to arrive more than ten years late). As such we made the decision to restrict all component lateness and buffering to a maximum lateness of one year, as we felt anything beyond this would be unrealistic in practice. For ease of computation, this was done by rounding any results greater than one year to be one year exactly. In the majority of cases this would have a limited impact, but in cases with high means or highly skewed distribution this could impact the results (Figure 21). This was deemed acceptable for three reasons: 1) such long delays are unrealistic because other suppliers or measures would be sought to ensure delivery continuity, 2) the extreme delays only occurred in very rare cases, and 3) in the cases where an extremely skewed distribution was clipped in this manner we were still able to determine that this distribution was highly skewed, minimizing the effect on our analysis. While the truncation of the lognormal distribution could have been done in a smoother way that spreads the

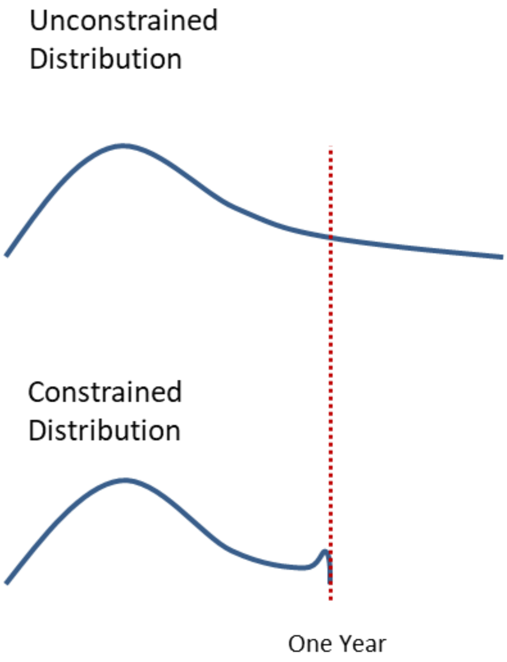


Figure 21 – The effect of clipping distributions

probability of the tail across its full domain, the problem of clipped distributions was limited to extreme cases and should not have any effect on our final conclusions.