Analyzing Controllable Factors Influencing Cycle Time Distribution in Semiconductor Industries

by

Tanushree Salvi

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved December 2016 by the
Graduate Supervisory Committee:

Jennifer Bekki, Co-Chair
Angela Sodemann, Co-Chair
Yi Ren
Abdelrahman Shuaib

ARIZONA STATE UNIVERSITY

May 2017

ABSTRACT

Semiconductor manufacturing is one of the most complex manufacturing systems in today's times. Since semiconductor industry is extremely consumer driven, market demands within this industry change rapidly. It is therefore very crucial for these industries to be able to predict cycle time very accurately in order to quote accurate delivery dates. Discrete Event Simulation (DES) models are often used to model these complex manufacturing systems in order to generate estimates of the cycle time distribution. However, building models and executing them consumes sufficient time and resources. The objective of this research is to determine the influence of input parameters on the cycle time distribution of a semiconductor or high volume electronics manufacturing system. This will help the decision makers to implement system changes to improve the predictability of their cycle time distribution without having to run simulation models. In order to understand how input parameters impact the cycle time, Design of Experiments (DOE) is performed. The response variables considered are the attributes of cycle time distribution which include the four moments and percentiles. The input to this DOE is the output from the simulation runs. Main effects, two-way and three-way interactions for these input variables are analyzed. The implications of these results to real world scenarios are explained which would help manufactures understand the effects of the interactions between the input factors on the estimates of cycle time distribution. The shape of the cycle time distributions is different for different types of systems. Also, DES requires substantial resources and time to run. In an effort to

generalize the results obtained in semiconductor manufacturing analysis, a non- complex system is considered.

ACKNOWLEDGMENTS

I would like to sincerely thank my advisor Dr. Jennifer M Bekki, for her constant support, encouragement and supervision throughout the masters' thesis. Her expert guidance, timely suggestions, meticulous scrutiny and approach has helped me in every step in accomplishing this work.

I would also like to thank Dr. Sodemann, Dr. Ren and Dr. Shuaib for serving in my committee and for their encouragement throughout the research. I would also like to thank all the faculty and staff of Mechanical and Aerospace Engineering department for their kind help and co-operation throughout my research and administrative work.

Finally, I would like to thank my family and friends and loved ones for their constant moral support and encouragement throughout my work.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

Table Page

Table | Page

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

1. Introduction

Semiconductor fabrication is one of the most complex manufacturing industries in today's global economy; manufacturers concentrate on producing high quality products at comparatively lower prices and at faster speeds. The industry is highly dynamic and competitive (Chang and Huang, 2002) and is largely driven by consumer satisfaction and rapidly changing consumer demand patterns. Cost and service are key influencers for customer satisfaction, and the on time delivery of finished products is considered to be an important factor in predicting service levels (Boyaci and Ray, 2006). Furthermore, given the rapid changes in customer demand patterns, effective planning in semiconductor manufacturing industry requires being able to not only predict customer delivery dates for current operating conditions but also to be able to accurately predict delivery times for future operating conditions (Yang et al., 2008).

The estimation of customer delivery dates, in turn, relies on having accurate estimates of cycle time, or the time required for a job to traverse a given routing in a production system (Hopp and Spearman, 2001). Typically, industries would like their cycle time to be as small as possible so that product can get to their customers as quickly as possible. Since not all parts that enter the system finish processing in the same amount of time, cycle time is also a random variable (Hopp and Spearman, 2001). Consequently, the cycle time for a particular product within a particular manufacturing facility can be described with the help of a cycle time distribution. There are many factors that influence the shape of the cycle time distribution. When the influencing controllable factors change, the shape of the cycle time distribution changes, in turn potentially influencing delivery date estimates provided to customers. The following section discusses the controllable factors that influence the cycle time in detail.

1. Controllable factors that influence the cycle time

The literature has called out a number of factors that affect the cycle time distribution of manufacturing system. Akcali et al., (2001), for example, point out that queuing time for equipment, waiting times due to preventive or breakdown maintenance, processing time, inspection time and transportation time all contribute to cycle time. Meidan et al. (2011) argue that the most common factors that influence cycle time are work in progress levels, bottlenecks in the system, rework rates, equipment down time, mean time between failures, mean times to repair, machine setup times, tool capacities, dispatching and scheduling policies. Sivakumar and Chong (2001) state that batch size, dispatching policies, material handling time, setup time and machine up time are some of the controllable factors that affect the cycle time and cycle time distribution. Factors such as work in progress (WIP) levels, queue length, product mix are not controllable whereas factors such as rework rates, mean time between failures, machine setup times etc. are controllable. The objective of this work is to determine the controllable factors that affect the cycle time.

Qi et al., (2002) and Chung and Huang (2002), both discuss how cycle time depends on batch processing. Qi et al., (2002) support their claims through an example in which a 50% reduction in the batch size created a nearly 10% reduction in cycle time. Siavkumar and Chong (2001) conducted studies to understand the relationship between cycle time spread and some controllable variables of the manufacturing system. They concluded that batching has the greatest impact on cycle time distribution in semiconductor backend manufacturing. A backend in semiconductor manufacturing is integrated circuit (IC) assembly and testing. According to their studies, and in-line with findings of Qi et al., (2002) the average cycle time increases with increasing batch size. They also studied the impact of machine setup time and time required to repair machine failures on the cycle time. Not surprisingly, they observed that reduction in time required to setup or time required to repair equipment failures reduces the cycle time.

Meidan et al. (2011) found that the number of operations that are carried by the bottleneck tool group in a semiconductor manufacturing fab is one of the key factors in estimating cycle time of the system. In a semiconductor manufacturing system, the quality of the products needs to be very high. In order to achieve the quality desired, a certain percentage of the processed parts undergo rework. Rework adds to the number of operations that are carried by a bottleneck tool group. Meidan et al. (2011) and Qi et al., (2002) concluded that re-entry of the part varies the queue length. Also, the processing time for rework is lesser than the initial processing time for the tool group. This variation in the processing time is an important factor in determining the cycle time.

Meidan et al. (2011) also stated that frequency of tool idleness increases the waiting time for operator or material. Cycle time consists of processing time, waiting time and transportation time. When the waiting time changes (increases or decreases), the cycle time changes. Hedman et al. (2013) and Meidan et al., (2011) observed that standard deviation of idle time of the tool affects the processing time. Operators may not readily take a lot off of a tool immediately after processing concludes. During this time, the tool is idle and the idle time cannot be recovered. Depending on the time the tool is idle, processing time can vary. Akcali et al. (2001) and Meidan et al. (2011) confirmed that waiting time for the parts in a semiconductor manufacturing system is dependent on tool availability.

Dispatching policies are job scheduling policies according to which jobs are prioritized for processing in a system. Ankenman (2011) and Akcali et al. (2001) concluded that dispatching policies are an important factor for determining the scheduling policies for the semiconductor fabrication. Akcali et al., (2001) also concluded that dispatching policies affect the cycle time of photolithography station in semiconductor manufacturing which is known to be a bottleneck tool. Dispatching policies are job scheduling policies according to which jobs are prioritized for processing in a system. Dispatching rules are useful for finding a reasonably good schedule with regard to a single objective such as the makespan, the total completion time, or the maximum lateness (Pinedo, 1995). Sivakumar and Chong (2001) concluded in their work that the change in dispatching policies affect the cycle time distribution. They found that FIFO (First in First Out) increased the 98th percentile as compared to ESD (Earliest Start Date).

Based on the literature reviewed in this section, the controllable factors that affect the cycle time considered in this work are listed in Table 1.

In order to have the complete understanding of the impact of change in controllable factors on the cycle time and its distribution, it is necessary to have a clear plan for how to describe the cycle time distribution. Section 2 describes the parameters of the cycle time distribution that are as a surrogate to describe it.

Table 1. Controllable factors influencing cycle time

#	Controllable factors influencing cycle time
1	Time Between Arrival (TBA)
2	Percentage of Rework
3	Mean Processing Time (MPT)
4	Coefficient of Variation (COV) at Unloading Operations
5	Mean Time Between Failures in Emergency Failures
6	Mean Time to Repair in Emergency Failures (EM)
7	Batch Size
8	Dispatching Policies

2. Describing the cycle time distribution

Often, an estimate of a distribution's mean or a combination of the mean and variance estimates are used to describe a distribution. However, these estimates alone are not sufficient for truly capturing the shape of a distribution. One approach to more accurately describing the complete shape of a distribution is to obtain estimates of the first four moments of the distribution. Mean, variance, skewness and kurtosis are these first, second, third and fourth moments respectively. As is widely known, the mean of a probability distribution is a measure of central tendency (Montgomery, 2014), while the variance is a measure of how tightly the data is clustered around the mean (Montgomery, 2014). Often, an estimate of standard deviation, which is the square root of variance, is used in place of the variance estimate. Skewness, the third moment of a distribution, measures the degree of asymmetry in a distribution around its mean (Green and Salkind, 2014). A symmetrical distribution, such as the normal distribution, has a skewness equal to 0, while a less symmetrical distribution, such as the exponential distribution, has a positive skewness. Kurtosis is a measure of the combined sizes of the two tails (Green and Salkind, 2014); distributions with greater probabilities associated with values that occur in the lower and/or higher extremes of the distribution have greater kurtosis. Kurtosis values are often compared to the kurtosis of the normal distribution, which is equal to 3. If the kurtosis of a particular distribution is greater than 3, then it is said to have "heavier tails" than a normal distribution.

Along with estimates of the first four moments, a set of percentile estimates are also able to provide a more complete picture of a distribution when compared to estimates of only the first two moments. Percentiles represent the percentage of observations that fall at or below the score of interest (Green and Salkind, 2014). A well selected set of percentile estimates can give a nearly complete picture of a distribution. For example, 90% of the products manufactured in a given facility have a cycle time that is equal to or less than the 90^{th} percentile of the cycle time distribution for that facility. In turn, quoting the 90^{th} percentile of cycle time as the delivery time would give the decision maker confidence that only 10% of the orders would be delivered later than the promised delivery date. Quoting smaller percentiles (i.e., 60^{th} percentile) as delivery dates yields shorter (and perhaps more desirable to the customer) delivery time estimates, but also is associated with greater risk (i.e., 40% of the orders would be expected to arrive after the promised delivery date). In the work here, estimates of the 25^{th}, 50^{th}, 75^{th} and 95^{th} percentiles are considered along with estimates of the first four sample moments describing the cycle time distribution.

In order to obtain the estimates of cycle time, semiconductor manufacturing companies typically develop models. Chang and Huang (2002) classified cycle time estimation into four categories: simulation, statistical analysis, analytical method and hybrid method. A simulation is the imitation of the operation of a real world process or system over time (Banks et al., 1996). Simulation models have been developed for capacity planning and cycle time reduction. For example, Wang and Wang (2007) developed a simulation model which can acquire optimal batch size to reduce cycle time under different bottleneck loading conditions. Chung et al. (2015) used simulations to show the advantages of a particular dispatching rule. Even before new system is deployed or changes to an existing system is made, simulation models can be created to predict the system's performance. It is faster and more cost effective than conducting experiments with the physical system (Yang et al., 2008). However, running simulation experiments requires extensive input data and substantial resources.

Statistical analysis can also be applied to determine the relationship between cycle time and other related parameters. Pearn et al., (2009), for example, considered a statistical approach for cycle time estimation incorporating the upper confidence bounds of estimated cycle times at various confidence coefficients in semiconductor plastic ball grid array packaging factories. Backus et al., (2006) applied a data mining approach to provide nonlinear predictor variables to estimate factory cycle time. They also pointed out a benefit of statistical methods in that the models themselves can be updated quickly (i.e., executing a statistical model is much less time-intensive than executing a simulation model).

Analytical methods for modeling cycle time are often based on queuing theory. Chung and Huang (2002), for example, provided an analytical approach to estimate cycle times for wafer fab with engineering lots. More recently, Shanthikumar et al. (2007) presented a survey for the application of queuing theory in semiconductor manufacturing systems. Analytical methods have shorter computational time and can be easily done when all the parameters of the system are known. However, analytical methods are often not as accurate as simulation for modeling high volume electronics manufacturing because the complex operational behaviors of these facilities cannot be represented adequately in a single analytical model.

Finally, the hybrid method combines different methods to produce a cycle time estimation. For example, the application of analytical methods and simulations in combination could be used to develop a dynamic cycle time estimation. Chang and Liao (2006) presented a flow-time prediction method which incorporates fuzzy rule bases with the aid of self-organizing map and genetic algorithm. In addition, Chen (2006) and Chen (2007) applied hybrid fuzzy c-mean and fuzzy back propagation network approaches to estimate cycle time in semiconductor manufacturing process.

Yang et al., (2008) argue that simulation is an essential tool for design and analysis of a complex manufacturing system such as a semiconductor fab. Moreover, Chung and Huang (2002) point out that simulation is an effective tool for analyzing and predicting the dynamic behavior of complex systems. Potential changes to the system can be first simulated in order to predict their impact on system performance. Simulation can also be used both as an analysis tool for predicting the effect of changes to existing systems, and as a design tool to predict the performance of new systems under varying set of circumstances.

3. Discrete Event Simulation (DES) modelling

Discrete event simulation is a particular type of simulation model in which the state variable changes only at a discrete set of points in time (vs continuously). DES models represent the stochastic and temporal behavior of a system and are a commonly used tool for modeling the operations of queueing based manufacturing systems. An advantage of DES is that it can be used to handle almost any level of system detail (Banks et al., 1996). Also of note is that the nature of DES is stochastic; since the input variables to the model are random variables, the output variables are also random variables.

A simple queueing system is portrayed in Figure 1. This system consists of a single machine. Parts arrive into the system with a particular arrival time. The job is then loaded to the machine with the help of an operator and is then processed. While a part is being processed, more parts arrive and a queue is built. When processing is finished, the part is unloaded and a new part from the queue is loaded on the machine for processing. This continues until all the parts are processed. A system such as this is typical for simulation in DES.

Semiconductor manufacturing systems are complex because of fluctuating demands, lot sizes, types of products flowing through the system, reentrant flow into the bottleneck tool, sequence dependent set up times, etc. (Meidan et al., 2011). Using DES the relationship between the controllable factors known to influence the cycle time distribution (described previously in this chapter) and the estimates of the parameters describing cycle time distribution (i.e. sample moments and percentiles) can be established.

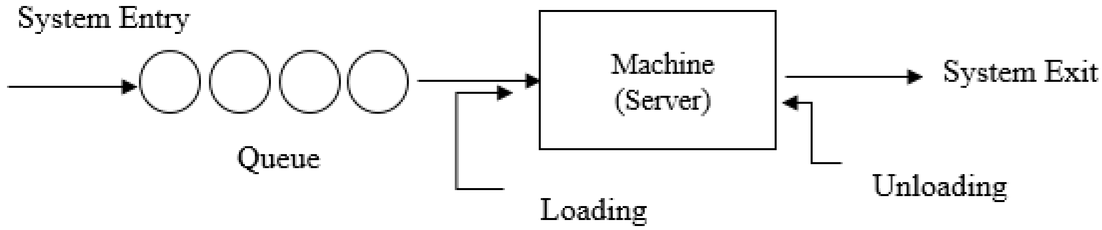


Figure 1. Simple queuing system

The effectiveness of DES modelling has also been shown to improve when combined with experimental design techniques. In this research, factorial experiments are used to analyze the effect of change in the controllable factors on a semiconductor manufacturing system. Section 4 elaborates on factorial experiments and highlights some literature in which DES and factorial designs are combined to analyze systems.

4. Factorial Experimental Design

Montgomery (2014) defines an experiment as a test or series of runs in which purposeful changes are made to the input variables of a system to observe and identify the reasons for changes in the output response. To do so, one needs to determine which input variables are responsible for the observed changes in the response, then develop a model relating the response to the important input variables and then use this model for system improvement or decision making.

In general, experiments are used to study the performance of processes and systems. The process or system can be represented by the model shown in Figure 2. The inputs to the system can be controllable factors or uncontrollable factors (noise). There may be a number of factors that affect a system. Montgomery (2014) states that factorial experiments are the best approach to deal with systems influenced by several factors. Also, Kumar and Nottestand (2006) stated that full factorial designs with simulations are preferred to analyze what-if questions since these designs provide the most information that is not aliased with main effects or two factor interactions. Answering what-if questions is a process of determining the effects on outcomes through systematic changes in the input. In manufacturing what-if questions involve answering questions for product mix, production targets and capital expansion (Yang et al., 2008).

In this research, DES is considered to be a process. The controllable factors mentioned earlier in section 1 are used as inputs to the simulations. The response variables or output measures are the estimates of the cycle time distribution.

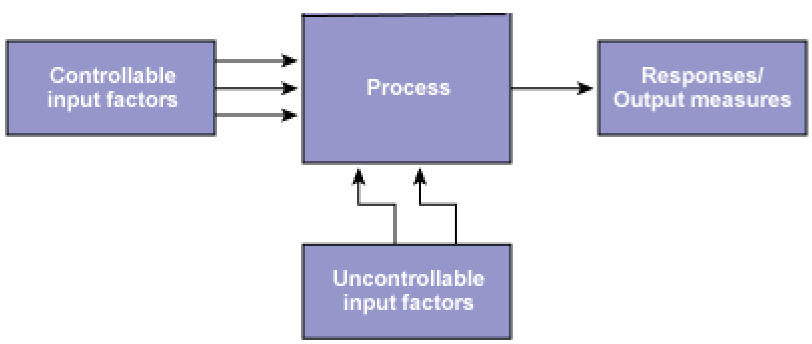


Figure 2. A process in DOE

A factorial experiment is an experimental strategy in which factors are varied together, instead of one at a time. A two level factorial design consists of two levels of each of the factor. If there are two factors say A and B, a two factor factorial experiment for studying the joint effects of these two factors on a response variable can be represented as shown in Figure 3. In this factorial experiment, both A and B have two levels (1 and 2) and all possible combinations of the two factors across their levels are used in the design. Geometrically, the four runs form the corners of the square in Figure 3. This particular type of factorial experiment is called 2X2 or 2^2 factorial design (two factors, each at two levels). Observations of the response variable are usually subjected to variation and uncertainty. Hence, replications are done for the factorial experiments to reduce the variability of the observations. This experimental design would enable the experimenter to investigate the individual effects of each factor and to determine whether the factors interact. Generally, if there are *k* factors, each at two levels, the factorial design would require 2^k runs.

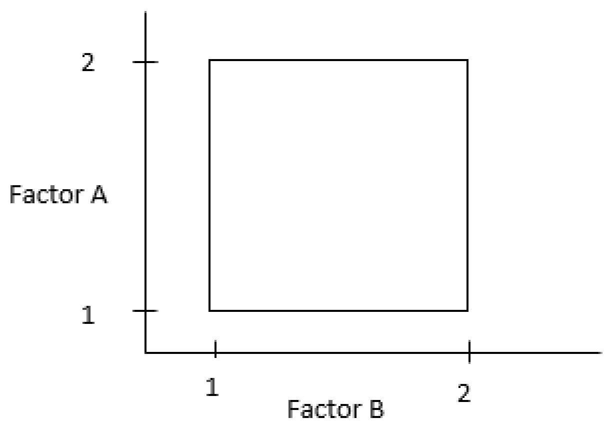


Figure 3. A two factor factorial experiment

Kumar and Nottestand (2006) stated that use of DOE with simulation allows a detailed understanding of a process to be obtained in a relatively short time. Furthermore, they concluded that the combination of simulation and designed experiments can bring projects to completion faster, with an improved outcome at a lower cost.

Also, Montevechi et al. (2010) presented a sensitivity analysis in DES models using factorial designs. They stressed that DOE approach (combination of simulation with factorial experimental design) improves understanding of the manufacturing system, generating further knowledge about the importance and significance of each resource used. They argued that despite of the relative time spent in the construction of the models, the twofold approach of DOE and simulation elucidates how the resource can be efficiently changed and employed. Finally, Yang et al., (2008) proposed a method for estimating multiple cycle time percentiles using DES and DOE.

As mentioned earlier, semiconductor manufacturing systems are dynamic and largely dependent on customer satisfaction making the simulation models to also alter dynamically. Rapid changes in demand make previously built simulations models outdated and irrelevant. Also, building simulation models and obtaining the necessary input data can consume a large amount of time and resources. Yang et al. (2008) argued that it may take several minutes or even several hours to complete simulation run for a given scenario of a semiconductor manufacturing facility. In order to conduct successful what-if analyses, simulation models should be run under all kinds of alternate configuration, making the time and effort required for the simulation analysis even more extreme.

The purpose of this research is to determine the influence of controllable input parameters on the cycle time distribution of a semiconductor or high volume electronics fabrication facility. This experimentation will provide important insights to decision makers about how the distribution changes. This will help manufactures to implement system changes to improve the predictability of parameters of the cycle time distribution (and, therefore, customer delivery dates) without having to actually utilize the resources required to execute simulation models. Rather than having to run multiple models to understand how change in factors changes the cycle time distribution, this work will help decision makers make similar inferences without having to build models and execute them.

This research will investigate the effects of the input factors of a semiconductor manufacturing systems, which were described earlier in this chapter, on the parameters describing the cycle time distribution (i.e., the first four sample moments and the 25^{th}, 50^{th}, 75^{th}, and 95^{th} percentiles). Worth noting is that the shape of the cycle time distribution is different for different systems. In an effort to understand the generalizability of the findings from the work in this thesis two types of system, one of significantly less complexity than the other, are examined. Mittler et al., (1995) present in their findings that normal distribution is a good approximation of the cycle time distribution for semiconductor fabrication facilities. However, this may not be true for manufacturing facilities that are not as complex as semiconductor manufacturing. If the results obtained from a complex system are valid to a system of less complexity, the results might hold across other systems bounded by the cycle time distribution of these two systems.

The remainder of this thesis will provide details about the two manufacturing systems used to conduct experiments, the methodology of the experiments (Chapter 2) and the results obtained (Chapter 3). Finally, discussions, conclusions and future work are presented in Chapter 4.

2. Experimental Plan and Methodology

The purpose of this research is to determine the influence of input parameters on the cycle time distribution of a semiconductor or high volume electronics fabrication. This experimentation will provide insights about how the distribution changes. This will help manufactures to implement system changes to improve the predictability of their cycle time distribution without having to run simulation models. With this understanding, they can make informed decisions about how changes to the production flow might influence the distribution and, relatedly, how delivery dates are quoted. In industries such as high volume electronics, where competition is heavily based on customer service where meeting delivery times is very crucial, this information will help decision makers in rapidly understanding how specific, potential changes to their production system might influence the delivery times that are quoted to customers.

The shape of the cycle time distributions is different for different types of systems. The presence of batching, the number of operations included in the production and the choice of dispatching policies (Sivakumar and Chong, 2001) all have an impact on the shape of the cycle time distribution. In general, with increasing complexity of the production environment, cycle time distributions tend to become closer to normal (Rose, 1999; Mittler, et al., 1995). Given the variety of potential distributional shapes and the interest in determining how generalizable conclusions from this research are, experimentation was done on two systems, one of significantly greater complexity than the other. For each system, the following overall process was followed during the course of this research:

1. A discrete event simulation model of the system was created and validated. Output from the model includes parameters describing the cycle time distribution.
2. Factors known to influence the cycle time distribution (and controllable on an actual production floor) were modified, and the DES model was executing under these varying conditions.
3. Analysis was conducted to understand how the changes to the identified factors relate to the recorded parameters of the cycle time distribution.

This chapter provides greater detail about each of these three steps. First, the systems themselves are described. The parameters selected to describe the cycle time distribution are then detailed, and the experimental plan for varying controllable factors in the models and then recording these parameters is detailed.

1. Description of the system

The two production facilities selected for experimentation in this work are a model of a simple job shop facility and a model of a more complex semiconductor manufacturing facility. The discrete event simulation models were built in a simulation software called ARENA, created by Rockwell Automation. A description of each of the two systems follows.

1.1 *Mini-Fab Model*

The Mini-Fab model is designed to capture in a simple format the key characteristics that make the modeling and scheduling of semiconductor manufacturing facilities particularly difficult: re-entrant flow, batching, set-ups, preventive maintenance, emergency maintenance and multiple part types. The model was designed by Intel and ASU (led by

Dr. Karl Kempf). This mini-fab model has also been used by other researchers (Chen et al., 2012).

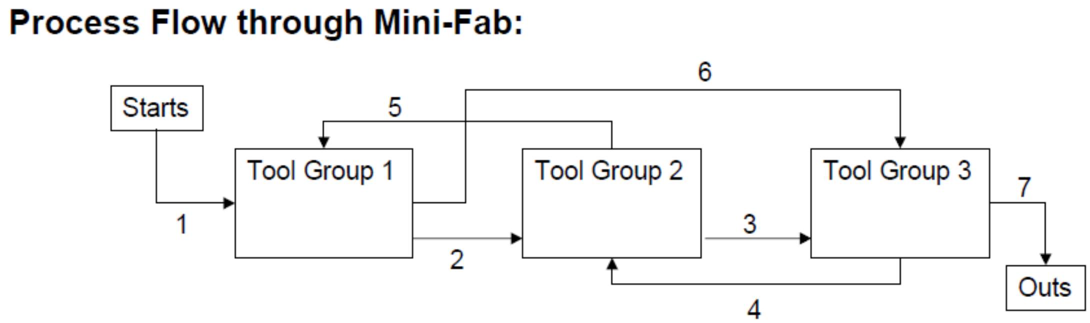


Figure 4. Product flow through mini-fab model.

There are two products that flow through the mini-fab model: Part X and Part Y. The distribution surrounding the time between arrivals for both part types follow an exponential distribution. For experimentation purposes, the time between arrivals (TBA) value for part type X can be changed. For part type Y it was set to EXPO (333.33) minutes. Figure 4 shows the product flow through Mini-Fab model.

Tool group 1 and 2 consist of two identical and parallel processing machines (Machine A and B for tool group 1 and Machine C and D with Tool group 2). Tool group 3 has just one machine (Machine E). There is one operator for each tool group and are utilized for loading, unloading and set-ups which serve as a secondary resource. The processing times at each station follow a normal distribution with a coefficient of variation (CoV) of 5% as shown in d for experimentation purposes.

Table 2 Machines in tool group 1 (similar to a diffusion oven in a semiconductor manufacturing system) perform batch processing similar to a diffusion oven in a semiconductor manufacturing system. After leaving the tool group, the batches are separated back into their original parts for the next processing step. Batches processed in step 1 can include part type X and part type Y but batches processed in step 5 cannot include different part types. Also, parts waiting for step 1 and 5 cannot be batched together. Of the parts completing processing at tool group 2 (similar to a photolithography stepper), 2% require rework. There is a single work station for rework and each rework operation takes 50% of the most recent processing time that the same job had at tool group 2.

Tool group 3 (e.g., an ion implanter) requires set-ups when changing between part types or between steps. Different sequences of parts require different lengths of set-ups. The set-ups are modeled with a normal distribution with CoV of 50%. When two sequential parts are to be processed on the same type, but are for different processing steps, the mean of setup time is 10 minutes. When two sequential parts are to be processed are of different type, but are for the same processing step, the mean of setup time is 5 minutes. Lastly, when the sequential parts are of different types and are to be processed are for different steps, the mean of setup time is 12 minutes. Tool group 3 also requires emergency maintenance. The time between failures is exponentially distributed with mean of 25 days and time to repair can be modeled with a gamma distribution with a scale parameter of 216 minutes and a shape parameter of 0.25. All the tool groups require preventive maintenance every 7 days and each session takes 1 hour of time. Also, all tool groups require condition checks every 30 days. Each condition check takes 6 hours of time. The dispatching policies in front of each machine is determined as FIFO (First In First Out) but can be changed for experimentation purposes.

Table 2. Characteristics of process steps in the mini- fab model

Process Steps	**Tool Group (Machine)**	**Processing Time (minutes)**	**Batch Size**	**Load Time (minutes)**	**Unload Time (minutes)**
1	1 (A, B)	NORM(225, 11.25)	3	NORM(20, 2)	NORM (40, 4)
2	2 (C, D)	NORM(30, 1.5)	1	NORM(15,1.5)	NORM(15,1.5)
3	3 (E)	NORM(55, 2.75)	1	NORM(10, 1)	NORM(10, 1)
4	2 (C, D)	NORM(50, 2.5)	1	NORM(15,1.5)	NORM(15,1.5)
5	1 (A, B)	NORM(225, 12.75)	3	NORM(20, 2)	NORM(40, 4)
6	3 (E)	NORM(10, 0.5)	1	NORM(10, 1)	NORM(10, 1)

1.2 *Job shop Model*

The second of the two systems that were modeled as an experimental vehicle for this research is a simple job shop. In job shop with m machines, each job has its own route to follow. The job shop is a model created by using model examples from Pinedo, 1995. There are three products that flow through the job shop, Type X, Type Y and Type Z. The distribution surrounding the time between arrivals for all the part types follows an exponential distribution, but the expected value of that distribution for part type X can be modified to control bottleneck tool utilization. For Part Type Y, Time between arrival (TBA) is set to EXPO (60) minutes and that for Part Type Z is EXPO (30) minutes. There are four tool groups in the model. Each tool group is made up of one identical, parallel machine and there is one operator for each tool group which serve as a resource. Figure 5 shows the product flow for Part Type X, Y and Z respectively through job shop model. Within the model, the dispatching policy can be changed to FIFO or SPT cat each of the stations.

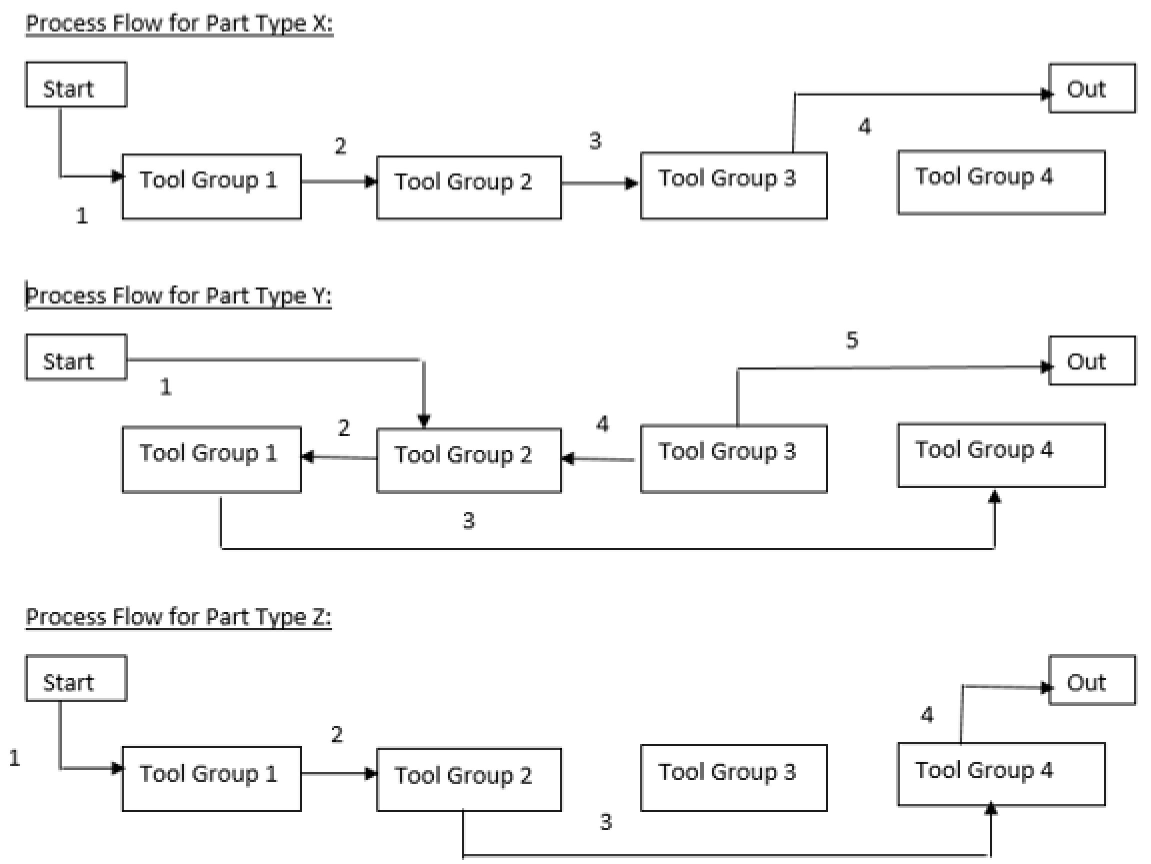


Figure 5. Product flow through job shop model.

Table 3, Table 4 and Table 5 shows the processing times for Part Type X, Y and Z respectively for job shop model.

Table 3. Processing times for Part Type X for the job shop model.

Process Step	**Tool Group**	**Processing Time (minutes)**
1	1	NORM (10, 0.5)
2	2	NORM (8, 0.4)
3	3	NORM (4, 0.2)

Table 4. Processing times for Part Type Y for the job shop model.

Process Step	Tool Group	Processing Time (minutes)
1	2	NORM (8, 0.4)
2	1	NORM (3, 0.15)
3	4	NORM (5, 0.25)
4	3	NORM (6, 0.3)

Table 5. Processing times for Part Type Z for the job shop model.

Process Step	Tool Group	Processing Time (minutes)
1	1	NORM (4, 0.2)
2	2	NORM (7, 0.35)
3	4	NORM (3, 0.15)

2. Special Considerations for Discrete Simulation Output Analysis

Due to the stochastic nature of discrete event simulation models, two executions of the same model with the same input parameter specifications will yield two different results. As a result, output measures (e.g., sample moment estimates and percentile estimates) from both the mini-fab and job shop models are also stochastic, random variables. Correspondingly, most statistical tests, including those that are conducted in this research, assume that these random variables are independent, identically distributed. However, these assumptions are not guaranteed to hold for random variables created as output measures from discrete event simulation models.

It is common to begin the execution of a discrete event simulation models using “empty and idle” conditions, in which all resources (i.e., tools, operators, etc.) are available, and all queues are empty. Such starting conditions make it so that jobs flowing through the model early in the simulation progress through the system more quickly than they would when the system is operating at steady state. As a result, estimates of parameters of the cycle time distribution from systems in which empty and idle starting conditions were employed can be biased (i.e., they can show that jobs progress through the system faster than they really should), violating the underlying statistical assumption that output measures as identically distributed. Such bias is referred to as initialization bias and is often addressed through the truncation of data acquired before the system reaches steady state operating conditions (Kelton, Sadowski R, Sadowski D, 2002). In the case of the two systems under investigation here, 250,000 time units of data and 30,000 time units of data at the beginning of every simulation run of the mini-fab model and job shop model respectively was truncated to remove initialization bias. These determinations were made based on a plot of time in system vs simulation run time from the most heavily utilized system configuration in each system.

The assumption of independence is also in question for discrete event simulation models of manufacturing systems. In such systems, the cycle time for a particular job is influenced by cycle times of jobs ahead of it. The cycle times are therefore not independent of each other. This is called auto-correlation (Kelton, Sadowski R, Sadowski D, 2002), and if not addressed, can cause variance estimates of parameter estimates to be artificially narrow. A common approach for dealing with this is to implement a lag between observations that are used in estimating parameters. If the lag value is large enough, then the values used in estimating parameters are approximately independent and not auto correlated. A lag of 2,000 and 25 between data points was utilized to generate pseudo independent and identically distributed (iid) observations for the mini-fab and job shop model respectively. The lag values for each system was determined based on a plot of auto-correlation vs. lag from the most heavily utilized system configuration.

3. Experimental Plan

A two level factorial design of experiments (DOE) was conducted to determine the impact of controllable factors on the parameters of interest that describe the cycle time distribution (i.e., estimates of the first four sample moments along with the selected percentile estimates). In the models of both the mini-fab and the job shop, each of the parameters describing the cycle time distribution was considered to be an independent response variable. As such, a separate experiment was conducted for each system on each of the response variables. The process for selecting the predictor variables, or those that are manipulated to investigate potential relationship to the response variables, for each system is described next.

3.1 *Mini-fab Predictor Variables*

Based on findings as described in Chapter 1, eight factors were considered for the mini-fab model. The factors, or predictor variables, are listed in the Table **6**6, along with their "baseline" value.

Table 6. Factors affecting cycle time of semiconductor fabrication.

#	Factors	Baseline values
1	Time Between Arrival (TBA) for Part X	EXPO (200) mins
2	Percentage of Rework	2%
3	Mean Processing Time (MPT)	Mean (Processing times from d for experimentation purposes. **Table** 2)
4	Coefficient of Variation (COV) at Unloading Operations	Mean (Unload time from d for experimentation purposes. **Table** 2)
5	Mean Time Between Failures in Emergency Failures	EXPO (25) days

6	Mean Time to Repair in Emergency Failures (EM)	GAMM (216, 0.25) mins
7	Batch Size	3
8	Dispatching Policies	FIFO

While each of these factors has been shown in the literature to generally influence the cycle time distribution, a pilot set of experiments was conducted to determine which had an impact on the cycle time distribution of the mini-fab system in particular. During these experiments, the mini-fab model was executed first with the predictor variables set to their "baseline" level. Then, the model was executed again, varying a single factor first increasing it by 10% and then decreasing it by 10%. The same factor was then increased by 30% and then decreased by 30%. Following the execution of each model, the percent change (from the baseline values) in the sample moment and percentile estimates of the cycle time distribution were recorded. The process was then repeated for each of the 8 candidate factors. Based on these pilot runs, factors that produced changes in at least two of the outcome measures (i.e., at least two sample moment and/or percentile estimates) by 10% were considered to be "sensitive" to the cycle time distribution.

Table 7 displays the final factors used in the factorial experiment, along with each of their levels. High and low levels for each factor were determined by the pilot runs. The goal was to test the mini-fab model for the most extreme utilization of the system. It was assumed that the extreme utilization for the mini-fab model would be if the levels of the factors were first increased by 30% from the baseline level (high level) and then decreased by 30% from the baseline level (low level). As an alternative to FIFO, SPT was considered to be as a high level for dispatching policies. The impact that changes to these factors visually on the cycle time distribution from the mini-fab model can be seen in Figure 6 to Figure 8. In each of these figures, the caption provides the levels for each of the factors in Table7. As illustrated, changes to these factors resulted in a cycle time distribution that looks approximately normal (i.e., Figure 6) to one that has much greater skewness (i.e., Figure 8). Quantifying the impact of these changes is a goal of the experimental design.

Table 7. Levels of factors considered in mini-fab model.

Factor #	Factor Name	High Level	Low Level
1	TBA for Part X (TBA)	EXPO (260) mins	EXPO (225) mins
2	Batch Size	10	3
3	Mean Processing Time (MPT)	+30% of mean	-30% of mean
4	Dispatching Policies(DP)	SPT	FIFO
5	COV of Unloading	50% of mean	10 % of mean
6	Mean time to Repair in EM (mean time to repair in EM)	280	150

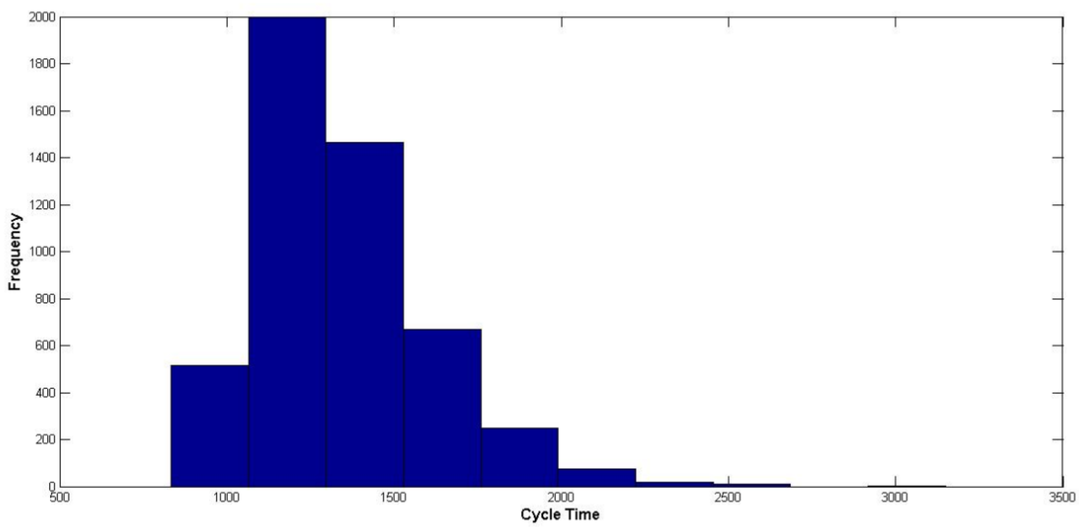


Figure 6. Histogram for run setting #2 for mini-fab model (TBA: high value, all other factors low level. Bottle neck utilization: 63.68%)

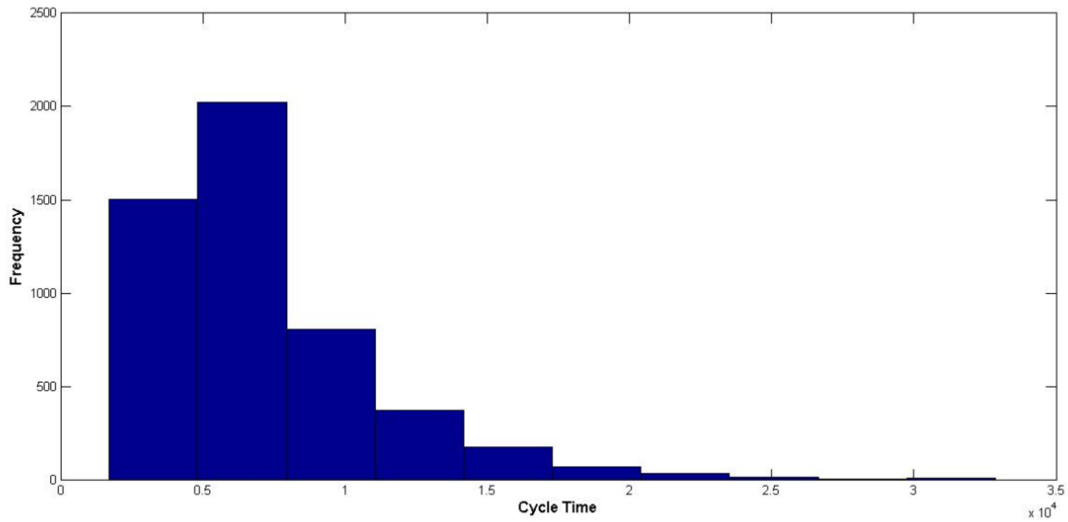


Figure 7. Histogram for run setting #39 for mini-fab model. (TBA, Dispatching Policies, COV: low value; batch size, MPT, Mean time to repair in EM: high value. Bottle neck utilization: 95.77%)

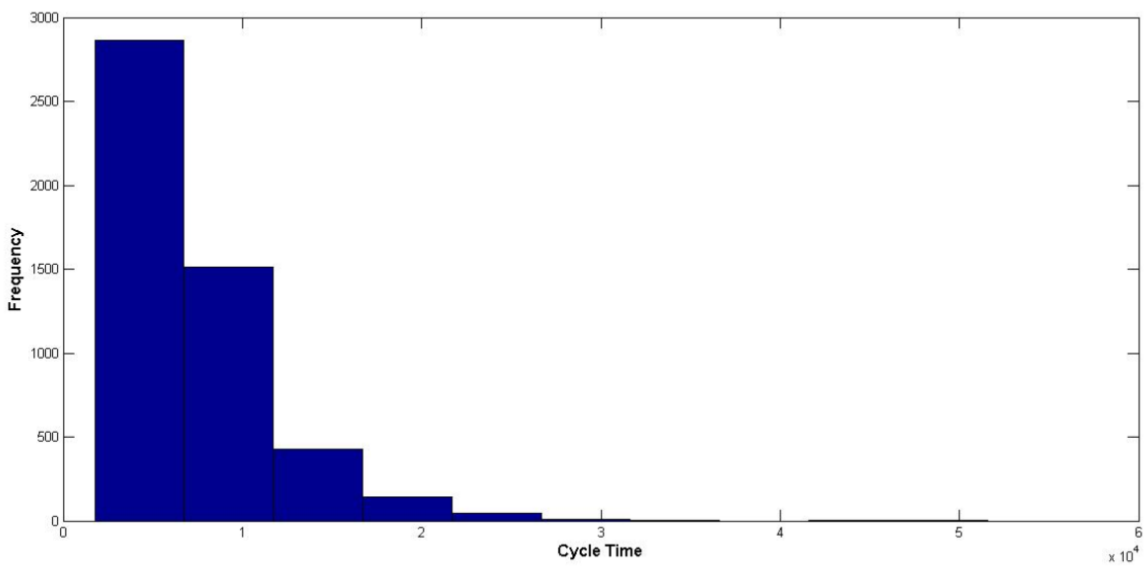


Figure 8. Histogram for run setting #55 for mini-fab model (TBA, dispatching policies: low value, batch size, MPT, COV, mean time to repair in EM: high value. Bottle neck utilization: 95.84%).

3.2 *Job shop Predictor Variables*

Of the 8 factors originally identified in the literature as influencing the cycle time distribution (see Table 6), only three were applicable to the job shop model: TBA for Part X, MPT and Dispatching policies. As a result, pilot runs were not conducted for job shop model. Instead, all three of the factors were simply included as factors in the experimental design. The factors and their levels are given in Table 8. The high and low levels for each factor were obtained by first increasing the levels of the factors by 30% from the baseline values (high level) and then decreasing the levels of the factors by 30% from the baseline values (low level). then decreasing the levels of the factors by 30% from the baseline values (low level). SPT was considered as a high level for dispatching policies against FIFO as a low level for dispatching policies.

Table 8. Levels of factors considered in job shop model.

Factor #	Factor Name	High Level	Low Level
1	TBA for Part X	EXPO (25.5)	EXPO (19.5)
2	Mean Processing Time	+30% of mean	-30% of mean
3	Dispatching Policies (DP)	SPT	FIFO

The impact that changes to these factors has visually on the cycle time distribution from the job shop model can be seen in Figure 9 to Figure 11. In each of these figures, the caption provides the levels for each of the factors in Table 8. As illustrated, and similar to what was observed with the mini-fab model, changes to these factors resulted in a cycle time distribution that looks closer to the normal distribution (i.e., Figure 9) to one that has much greater skewness (i.e., Figure 11). Quantifying the impact of these changes is a goal of the experimental design.

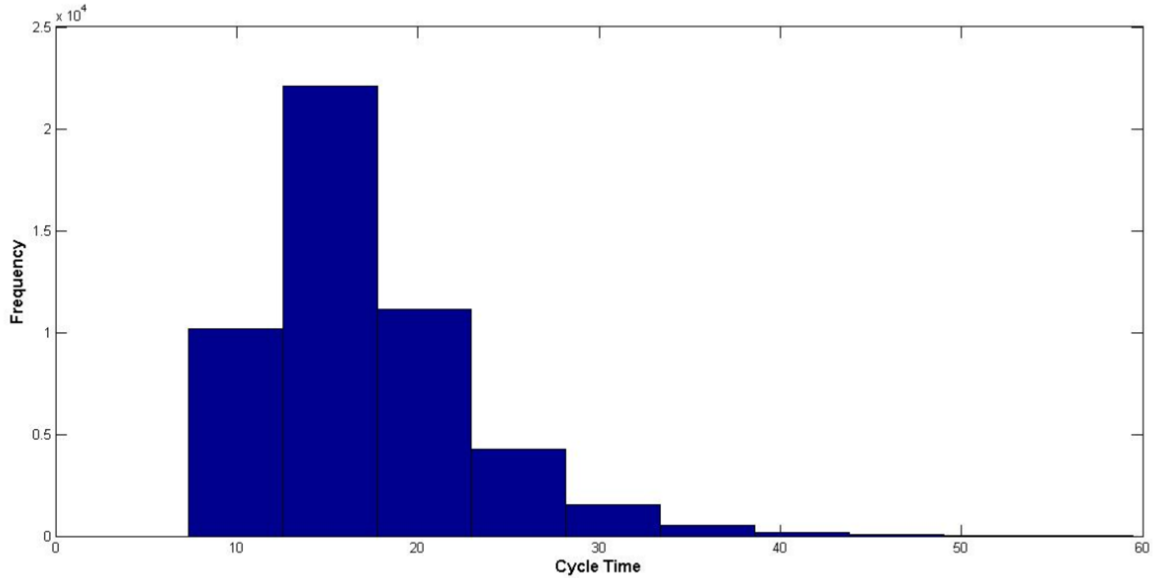


Figure 9. Histogram for sun setting # 2 for job shop model (TBA: high value, MPT, dispatching policies: low value. Bottle neck utilization: 47.63%).

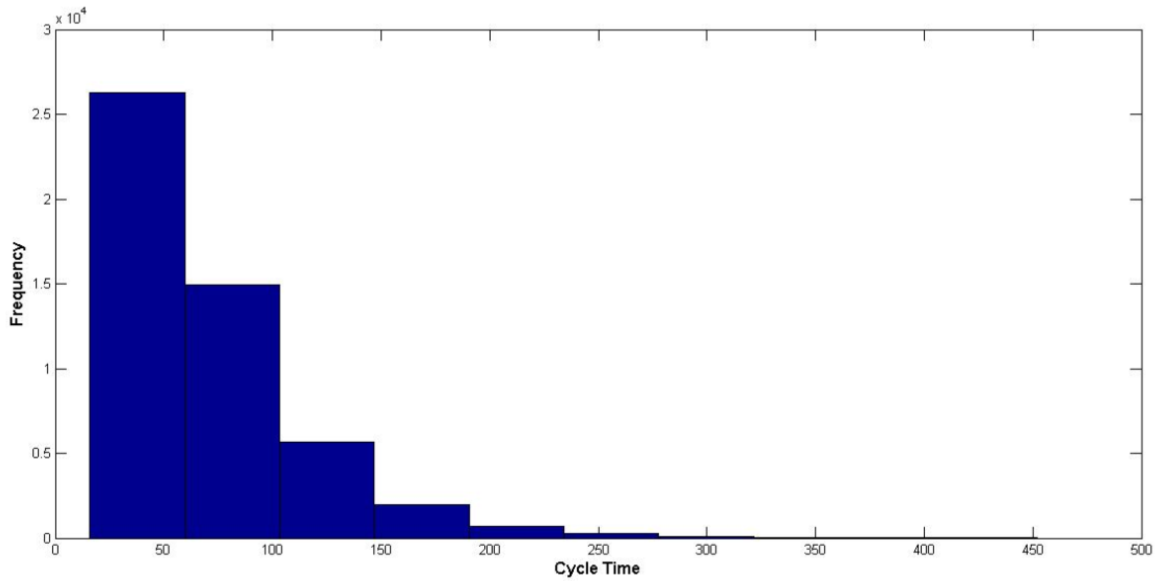


Figure 10. Histogram for sun setting # 4 for job-shop model (TBA: high value, MPT: low value, dispatching policies: high value. Bottle neck utilization: 88.43%).

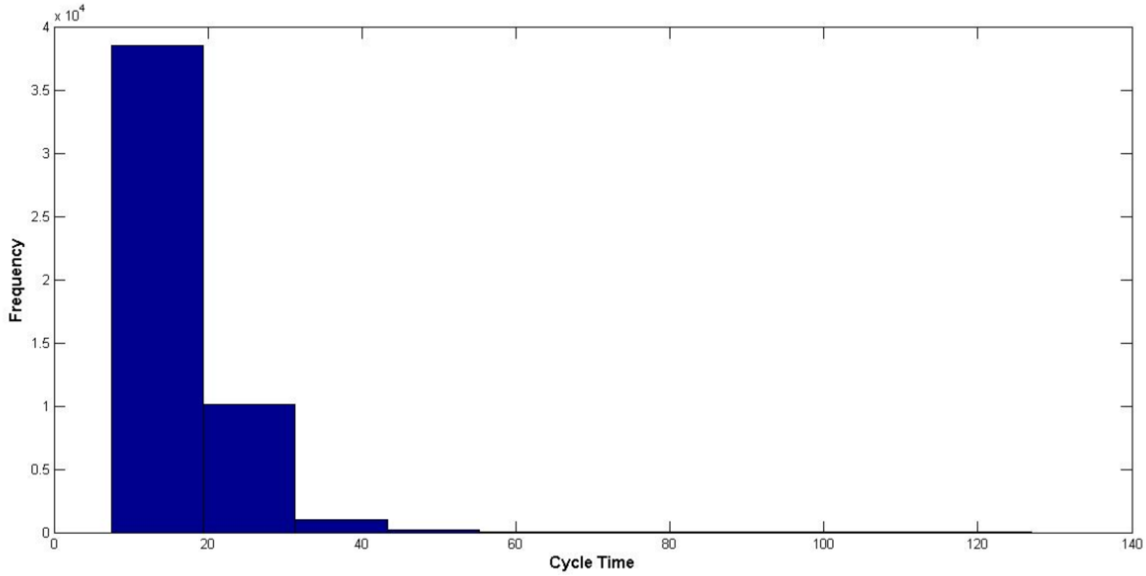


Figure 11. Histogram for sun setting # 6 for job shop model (TBA: high value, MPT: low value, dispatching policy: high value. Bottle neck utilization: 47.63%).

The underlying assumptions of factorial DOE are: 1) that the observations must be independent 2) that each factor used in DOE are chosen by the experimenter 3) that the design is completely randomized, 4) that the response variable follows a normal distribution, and 5) that the variance of the data points should be equal (Montgomery, 2014). While the assumption regarding iid data is met through the adjustments to the simulation output data describe previously, the assumption regarding the normality of the output measures is not. Specifically, with the exception of the sample mean estimates, the distributions describing the sample moments and percentiles obtained from simulation models are not normal. In order to satisfy the normality assumption, inferences from the central limit theorem were first employed used. The central limit theorem (CLT) states that, given certain conditions, the arithmetic mean of a sufficiently large number of observations of an independent random variable (i.e., in our case the sample moment and percentile estimates), will be approximately normally distributed, regardless of the underlying distribution (Montgomery, 2014). Carrying this idea forward to the output from the job-shop and mini-fab models implies that, while the distributions of the sample moments and percentile estimates themselves may not be normally distributed, the distribution surrounding sample means of these same variables do follow a normal distribution. Thus, for mini-fab model, 12 simulation runs were made to obtain 2 independent replications at each design point (i.e., the sample mean of six observations of each output measure were used as the response variable).

To evaluate the effectiveness of this approach in inducing normality in the output variables, a normal quantile plot was generated. In such a plot, if the underlying distribution is normal, the points on the plot will approximately follow a straight line. However, as illustrated in Figure 12, even after employing inferences from the CLT, the points were still not normally distributed. The CLT is known to be most effective when it is based on the average of a large sample of data (vs. the 6 data points that were averaged for this work). As such, additional effort was required to meet the normality assumption of factorial DOE.

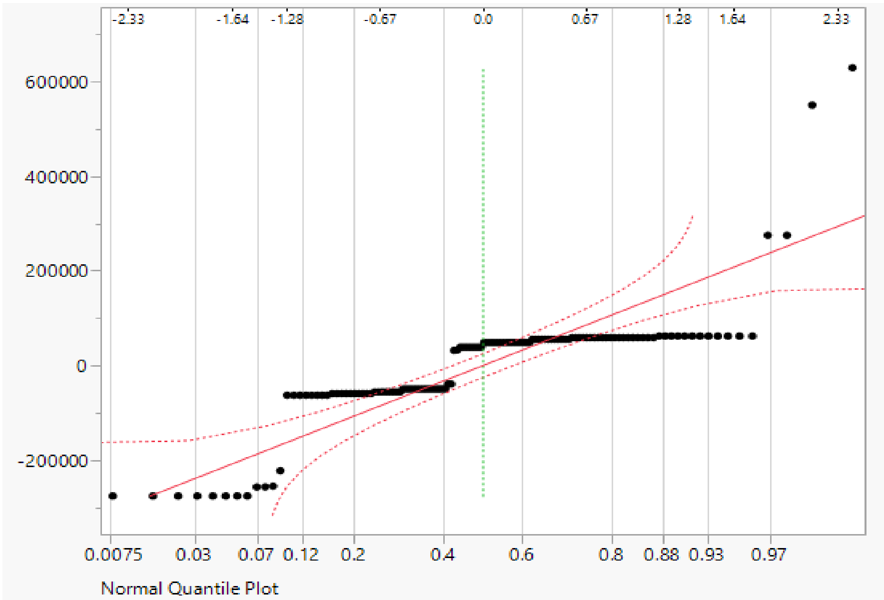


Figure 12. Normal quantile plot of residual of non- transformed data for 25th percentile of cycle time distribution for mini-fab model.

Transforming the data is often helpful in bringing a distribution closer to the normal distribution (Montgomery, 2014). The most common transformations used are a square root transformation, a logarithmic transformation and an inverse transformation. Each of these transformation was attempted, and the logarithmic transformation was found to have the best fit across all the output variables. Figure 13 illustrates the normal quantile plot the logarithmic transformed data for 25th percentile of cycle time distribution for mini-fab model. Compared to Figure 12, all the data now was within the 95% confidence interval of the straight line, hence making the data normal.

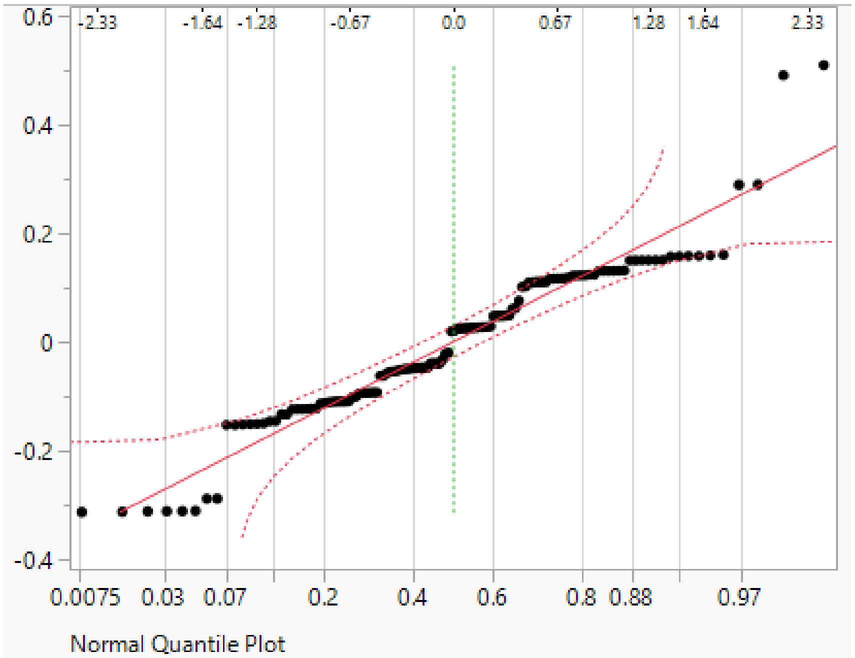


Figure 13. Normal quantile plot of residual of logarithmic transformed data for 25th percentile of cycle time distribution for mini-fab model.

Similar to the mini-fab model, for job shop model, the data was transformed using logarithmic transformation, square root transformation and inverse transformation. Among these transformations, the logarithmic transformation was again considered to be the most effective. Figure 4 and Figure 5 illustrate the normal quantile plot of non-transformed data and of logarithmic transformed data for the skewness of cycle time distribution for job shop model. It is notable that the same transformation worked for both systems. As seen from Figures 6 and 11, which illustrate the cycle time distribution for approximately the same utilization levels in each of the two systems, the shape of the cycle time distribution across both systems is not the same. The fact that the same transformation worked for both systems gives an indication of the robustness of the procedure.

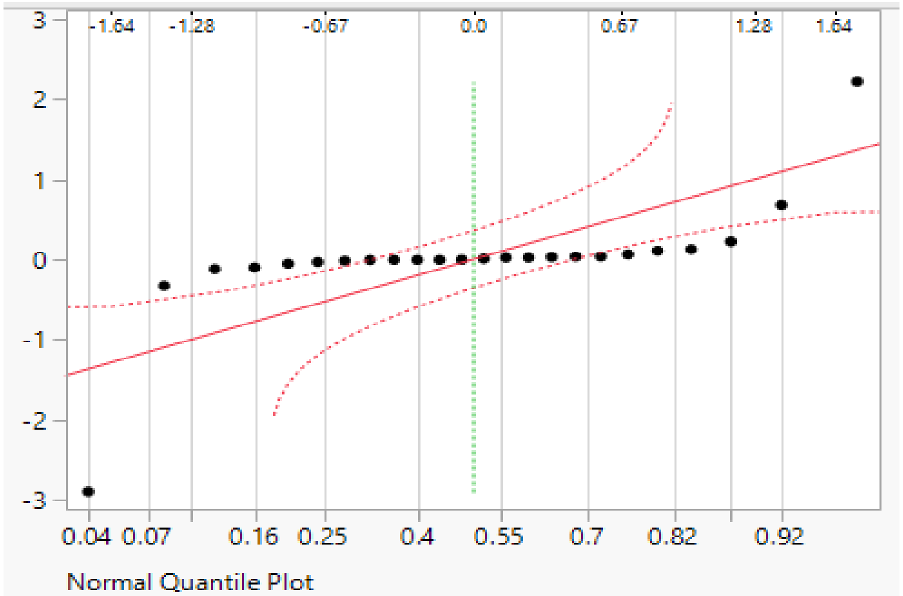


Figure 14. Normal quantile plot of residual of non-transformed data for skewness of cycle time distribution for job shop model.

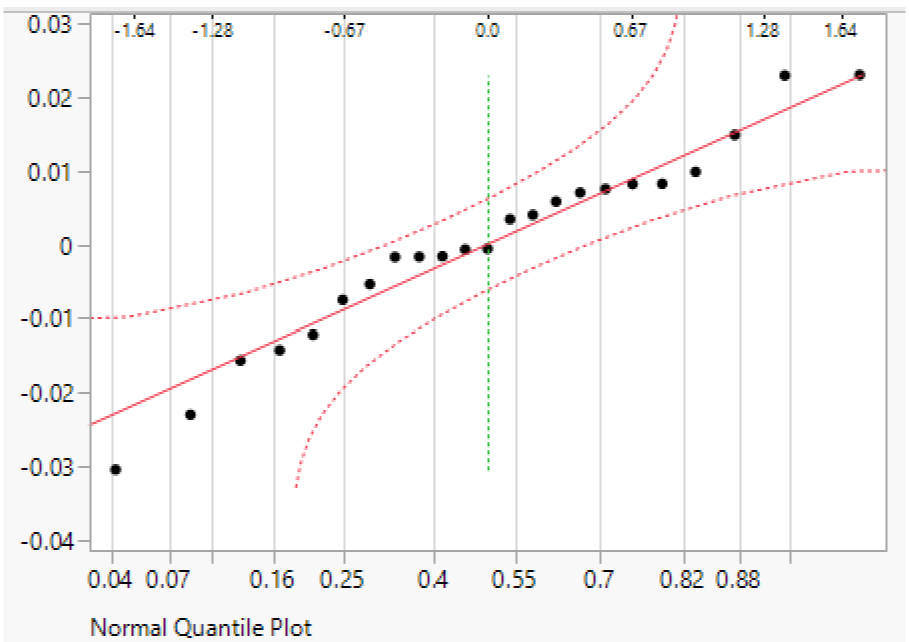


Figure 15. Normal quantile plot of residual of logarithmic transformed data for skewness of cycle time distribution for job shop model.

Based on the discussion above, Table 9 shows the summary of methodology for conducting experiments on mini-fab model and job shop model.

4. Experimental Plan Summary

The systems used are mini-fab model and job shop model. The factors sensitive to the mini-fab model were identified by pilot runs for analyzing sensitivity. Assuming extreme utilizations of the system, the levels of the factors were chosen. All the factors in the job shop model were considered and the levels for these factors were chosen in a similar way as for the mini-fab model. The simulation run length was 5,000 pseudo i.i.d. observations for mini-fab model and 50,000 pseudo iid observations for job-shop model. For conducting experiments, six independent simulations were run for the mini-fab model and three independent simulations were run for the job shop model. Two complete experimental replications were obtained for the mini-fab model, and three complete experimental replications were obtained for the job-shop model. The output from the simulation model (i.e. the cycle times) where the input to a MATLAB code to generate the estimates of the cycle time distribution (i.e. mean, standard deviation, etc.). A 2- level full factor DOE was conducted to see the impact of changes made to the factors on the shape of the cycle time distribution. To meet the normality assumption for DOE, the data points were transformed using a logarithmic transformation. All the analyzes for the mini-fab model and the job shop model explained in the following chapter were conducted using the transformed data. Figure 16 illustrates the summary of experimental plan conducted in this research. Table 9 shows the summary of the characteristics of DES output analysis for both models.

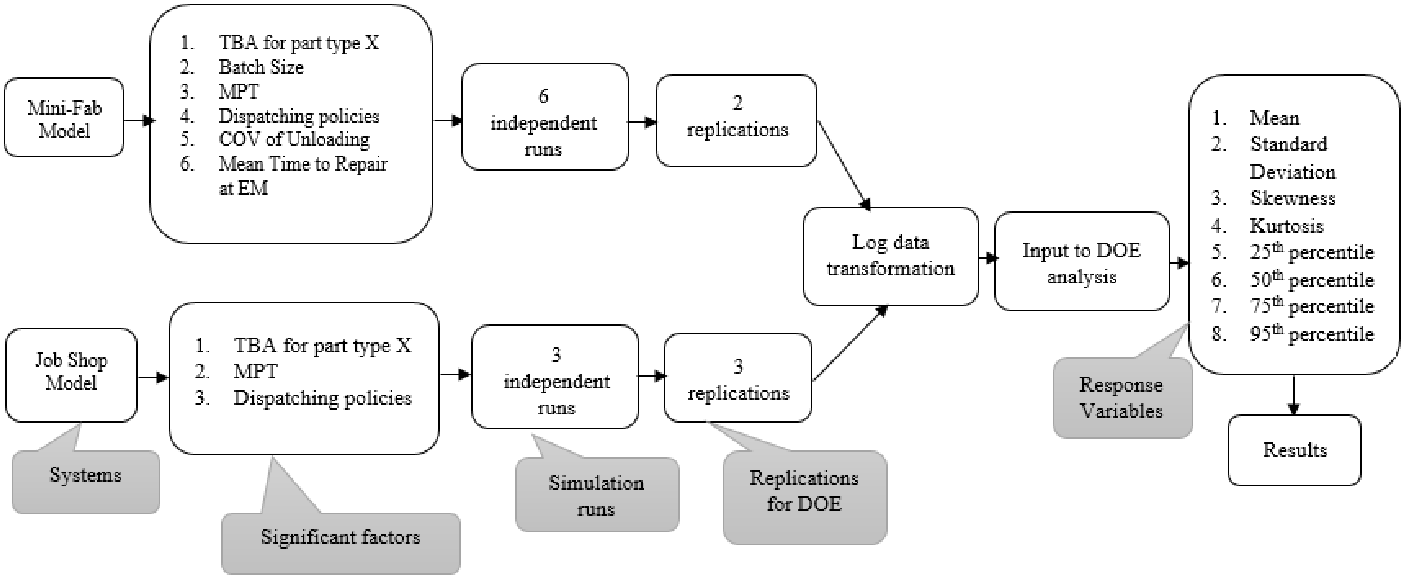


Figure 16. Summary of experimental plan

Table 9. Summary of characteristics of DES output analysis

Parameters	Mini-Fab Model	Job Shop Model
Simulation Run Length	5000 pseudo iid	50000 pseudo iid
Number of replications for DOE	2	3
Number of replications for simulation	12	9
Truncation of initial data	250000 time units	30000 time units
Lag Value	2000	25

3. Results

The objective of this research is to understand how changes to controllable factors on a production floor affect specific attributes of the cycle time distribution. This analysis will enable decision makers to have an increased understanding of how changes under consideration for the manufacturing system can be expected to change the cycle time distribution, in turn allowing improved delivery date quotations to be made without incurring the burden of additional simulation modeling and analysis. As mentioned in Chapter 2, a 2^6 full factorial DOE was conducted. In the DOE, factors included controllable aspects of the production system for which the cycle time distribution was found to be sensitive, and the response variables are attributes of the cycle time distribution (mean, standard deviation, skewness, kurtosis and percentiles). The levels of factors are shown in Table 7. In the interest in determining how generalizable the conclusions from the mini-fab model are, a 2^3 full factorial DOE of job shop model was also conducted to analyze if the results in the mini-fab model are also valid to the job shop model.

This chapter provides the results from conducting the experiments described in Chapter 2. Results for the mini-fab system are presented first, and results for the job shop model follow. For each system, main effects, along with two and three way interactions between factors that were found to be statistically significant are identified and described using p-values and effect sizes. Interactions between factors are further investigated to understand the nature of the interactions and their impact on the response variable(s). For each system (i.e., mini-fab and job shop), some main effects and interactions were found to be statistically significant in influencing all eight of the response variables (sample moment estimates and percentiles), while some factors were only influential to a subset of the response variables. Within the results for each model, factors that influenced all response variables are described first, as their cross-cutting influence is significant to understanding the cycle time distribution; a description of factors that influence only a subset of the response variables then follows.

1. Results for mini-fab model

The first objective was to determine which main effects, two- way and three-way interactions are significant in predicting each response variable in the mini-fab model. Table 10 shows the main effects, two and three way interactions of factors that were statistically significant ($p< 0.05$) across all response variables (the two-way interaction, for example, TBA*MPT is read as the interaction between TBA and MPT. The three-way interaction, for example, TBA*MPT*dispatching policies is read as the interaction between TBA, MPT and dispatching policies). In other words, these factors were found to statistically impact *all* of the eight response variables that describe the cycle time distribution: mean, standard deviation, skewness, kurtosis, along with the 25th, 50th, 75th, and 95th percentiles.

The first step in understanding the impact of the main effects and interactions on the response variable is to determine statistical significance of the factor(s) and to look at the magnitude and direction of the effect sizes. The magnitude of an effect size measures the expected change in the response variable per unit change in the factor(s). The direction of the effect size is an indication of whether the response variable increases or decreases with respect to the factors and their interactions. For example, if the effect size is -3 for an input factor with two levels, it means that the response variable decreases by a value of 3 per unit change in the input factor. For ease in analyzing the results, coded effect sizes are used. The coding of effect sizes positions them all on common scale, making it easier to compare the relative sizes and corresponding impacts on a response variable.

The effect size and p-values for each significant main effects and interactions are shown in Table 10 and Table 11 respectively. The effect sizes range in magnitude from 0.02 to 0.6. Following these tables, results are provided first for the main effects found to be statistically significant across all the response variables and then for the interactions that were found to be statistically significant across all response variables.

Section 1.1 explains the main effects common across all the response variables, including how each of the main effect affects the response variables.

1.1 ***Main Effects for Mini-fab model common across all response variables***

Three main effects were found to be statistically significant in predicting all the response variables: TBA, MPT, and DP. Figure 17 summarizes these main effects for each of the response variables. It shows that the TBA factor affects the standard deviation of the cycle time distribution the most and the skewness of the cycle time distribution the least. Changing MPT has a strong influence on the standard deviation of the cycle time distribution and a weaker influence the on 25^{th} percentile of the cycle time distribution. Finally, changes in dispatching policies have the most effect on kurtosis and have the least effect on the standard deviation of the cycle time distribution.

Table 10. Effect size values for all significant main effects and interactions for all performance measure for mini-fab model.

	Effect Size							
	Mean	**Standard Deviation**	**Skewness**	**Kurtosis**	**25th Percentile**	**50th Percentile**	**75th Percentile**	**95th Percentile**
Main Effects and interactions								
TBA	-0.199	-0.346	-0.053	-0.103	-0.121	-0.141	-0.160	-0.219
MPT	0.320	0.606	0.299	0.440	0.220	0.245	0.276	0.378
Dispatching Policies	-0.098	0.032	0.252	0.420	-0.129	-0.151	-0.162	-0.108
TBA* MPT	-0.210	-0.367	-0.025	-0.055	-0.129	-0.152	-0.174	-0.234
Batch Size* MPT	-0.198	-0.320	0.049	0.026	-0.129	-0.152	-0.173	-0.217
Batch Size* Dispatching Policies	-0.102	0.025	0.231	0.370	-0.133	-0.154	-0.165	-0.111
MPT* Dispatching Policies	0.082	0.034	-0.075	-0.105	0.108	0.122	0.123	0.093
TBA* Batch Size* MPT	0.160	0.176	-0.055	-0.042	0.114	0.127	0.133	0.141
TBA* Batch Size* Dispatching Policies	-0.084	-0.052	0.071	0.084	-0.111	0.123	-0.119	-0.098
Batch Size* MPT* Dispatching Policies	0.084	0.046	-0.029	-0.047	0.110	-0.125	0.127	0.099

Table 11. p- values for all significant main effects and interactions for all performance measure for mini-fab model.

	p- values							
	Mean	**Standard Deviation**	**Skewness**	**Kurtosis**	**25th Percentile**	**50th Percentile**	**75th Percentile**	**95th Percentile**
Main Effects and interactions								
TBA	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
MPT	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Dispatching Policies	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
TBA* MPT	0.000	0.000	0.027	0.000	0.000	0.000	0.000	0.000
Batch Size* MPT	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000
Batch Size* Dispatching Policies	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
MPT* Dispatching Policies	0.000	0.03	0.000	0.000	0.000	0.000	0.000	0.000
TBA* Batch Size* MPT	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
TBA* Batch Size* Dispatching Policies	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Batch Size* MPT* Dispatching Policies	0.000	0.000	0.010	0.000	0.000	0.000	0.000	0.000

	Mean	Standard Deviation	Skewness	Kurtosis	25th Percentile	50th Percentile	75th Percentile	95th Percentile
TBA	-0.199	-0.346	-0.053	-0.103	-0.121	-0.141	-0.160	-0.219
MPT	0.320	0.606	0.300	0.440	0.220	0.245	0.276	0.378
DP	-0.098	0.032	0.252	0.420	-0.129	-0.151	-0.162	-0.108

Figure 17. Main effects of TBA, MPT and dispatching policies on all response variables for the mini-fab model.

Table 10 illustrates that the effect size of the TBA factor for all response variables is negative. This implies that when the TBA increases from a lower value to a higher value the magnitude of each response variable decreases. This is not surprising given that as the time between entity arrivals to the system increases, fewer entities overall are present in the system, making the queue shorter. Figure 18 and Figure 19 show the main effects plot of TBA on mean and kurtosis of cycle time distribution respectively. The main effects plot examines differences between level means for the factor graphing the response mean for each factor level and connecting these values with a line. Figures 18 and 19 illustrate that the slope of the line connecting the two levels of TBA for mean of the cycle time distribution is greater than the slope of the line connecting the two levels of TBA for the response variable of kurtosis. This is also illustrated by the effect sizes for the TBA factor in Table 10.

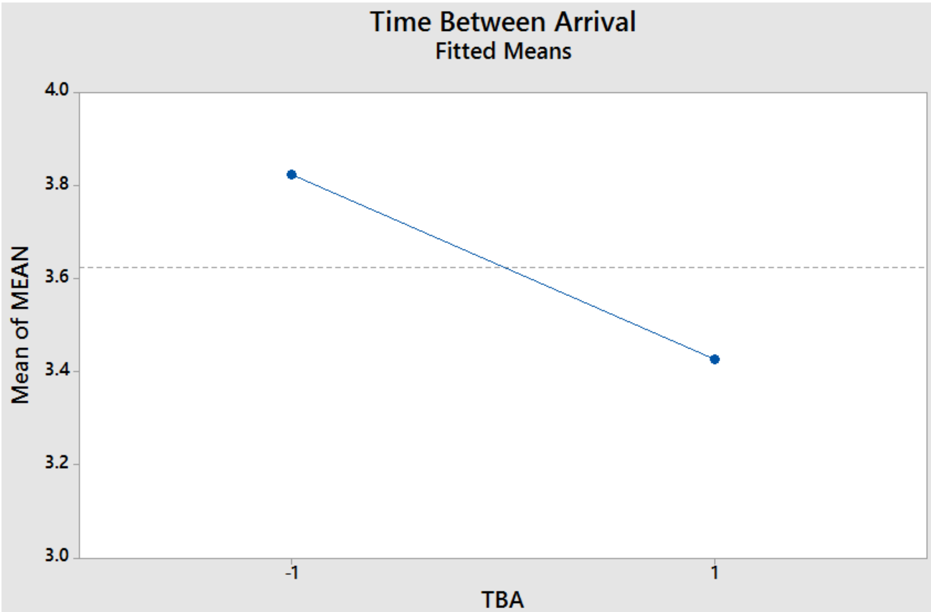


Figure 18. Main Effects plot of TBA on mean of cycle time distribution for the mini-fab model.

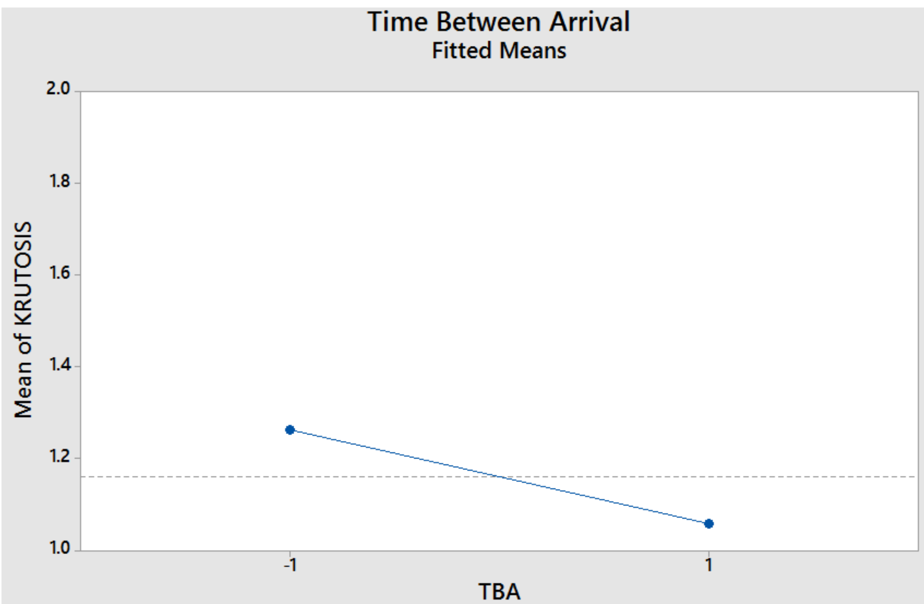


Figure 19. Main Effects plot of TBA on kurtosis of cycle time distribution for the mini-fab model.

Table 10 shows that the effect size of MPT for all response variables is positive, indicating that as MPT increases from a lower value to a higher value, the value of each of the response variable of the cycle time distribution increases. As the MPT increases, it follows directly that each station takes longer to process a job, thus increasing the mean and percentiles of cycle time distribution. As each entity takes a longer time to process at a station, the time between arrivals to each subsequent station also increases, in turn increasing the standard deviation of the cycle time distribution. Figure 20 and Figure 21 show the main effects plot of MPT on standard deviation and the 25th percentile of cycle time distribution respectively. As seen from Figure 20 and Figure 21, the slope of line connecting the levels the levels of MPT for standard deviation of cycle time distribution (0.608) is greater than that for the 25th percentile of cycle time distribution (0.220). This indicates that MPT has a greater impact on standard deviation than 25th percentile of cycle time distribution.

Dispatching policies were found to affect the mean and the percentiles of cycle time distribution in the mini-fab model differently than the standard deviation, skewness and kurtosis of cycle time distribution. Figure 22 and Figure 23 illustrate the main effects plot of dispatching policies on mean and skewness of the cycle time distribution. As the dispatching policies change from FIFO to SPT, the mean and percentiles of the cycle time distribution decreases. On the contrary, as the dispatching policies change from FIFO to SPT, the standard deviation, skewness and kurtosis of the cycle time distribution increases. When the SPT dispatching rule is employed, the queue is ordered to prioritize jobs with shorter processing times of the tool. So, whenever a machine is freed, the entity in the queue with shortest processing time begins processing next. This in turn, means that entities with longer processing times are disadvantaged compared to FIFO dispatching policies, and may take a very long time to exit the system (Pinedo,1995). The result of this is that the mean and the percentiles of the cycle time distribution decrease but the skewness and kurtosis of the cycle time distribution increases. Section 1.2 explains how the two-way interactions that are common across all the response variables affect the response variables.

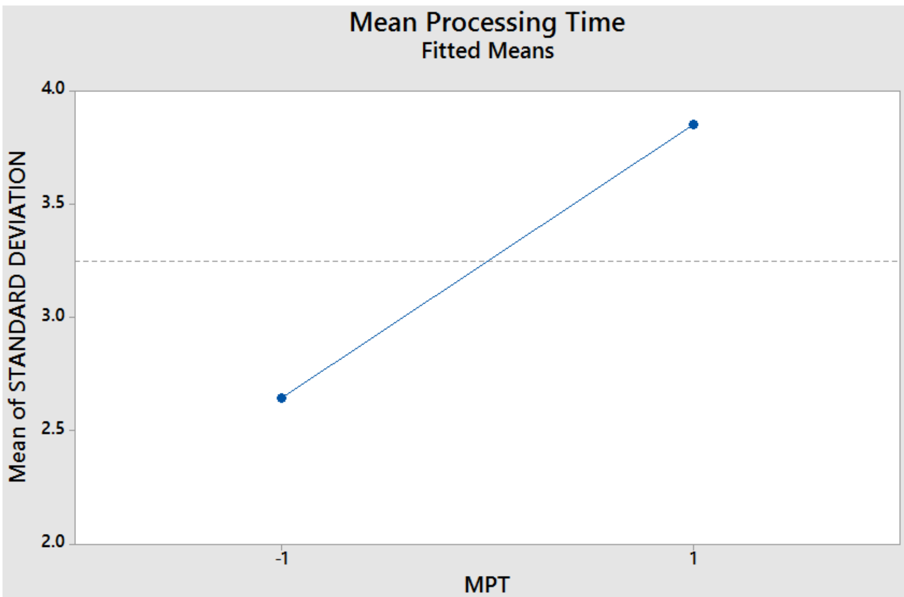


Figure 20. Main Effects plot of MPT on standard deviation of cycle time distribution for the mini-fab model.

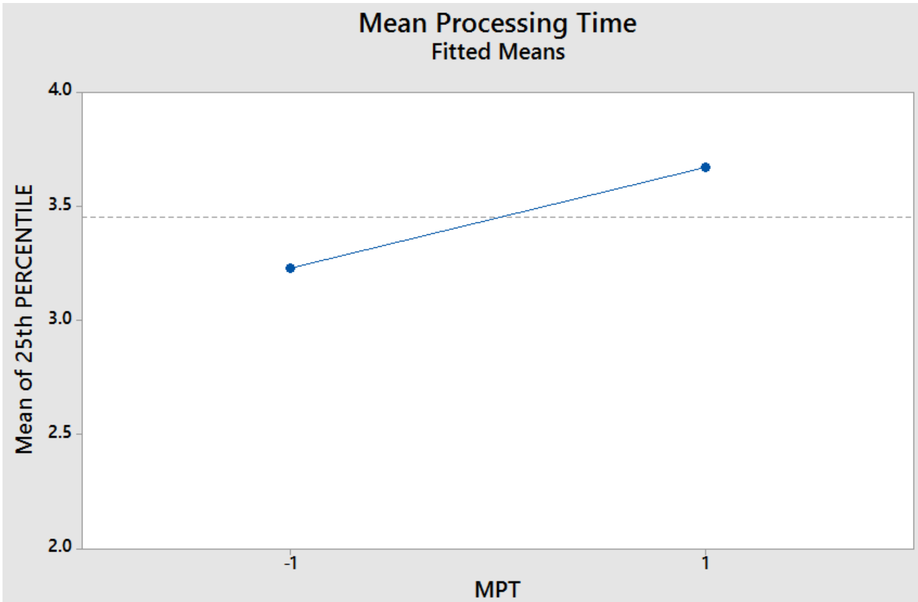


Figure 21. Main Effects plot of MPT on 25th percentile of cycle time distribution for the mini-fab model.

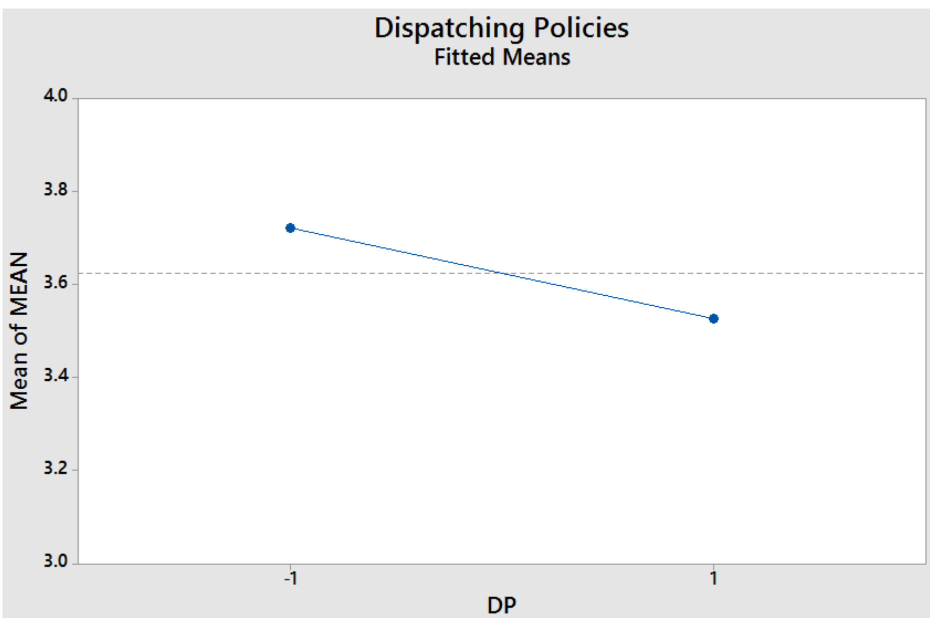


Figure 22. Main Effects plot of dispatching policies on mean of cycle time distribution for the mini-fab model.

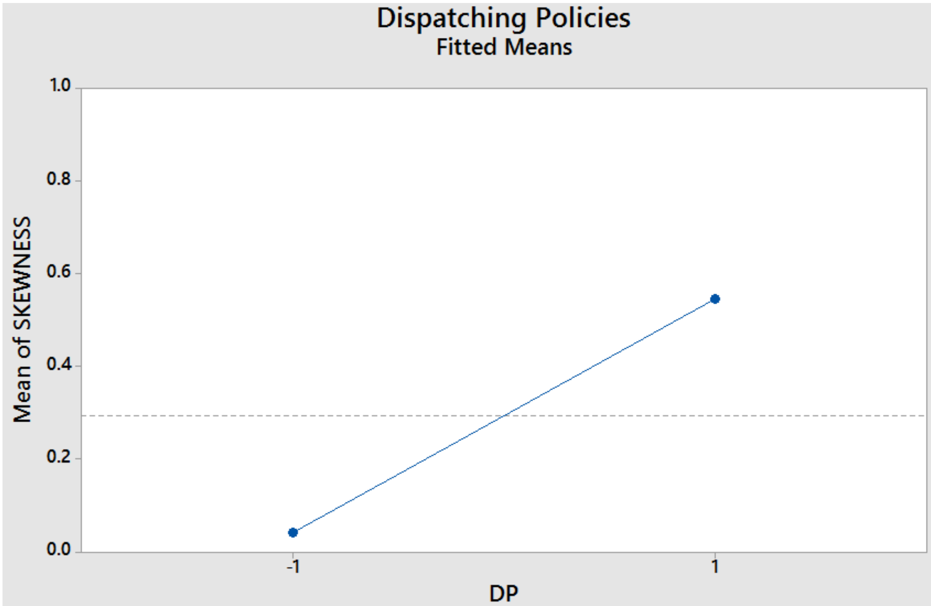
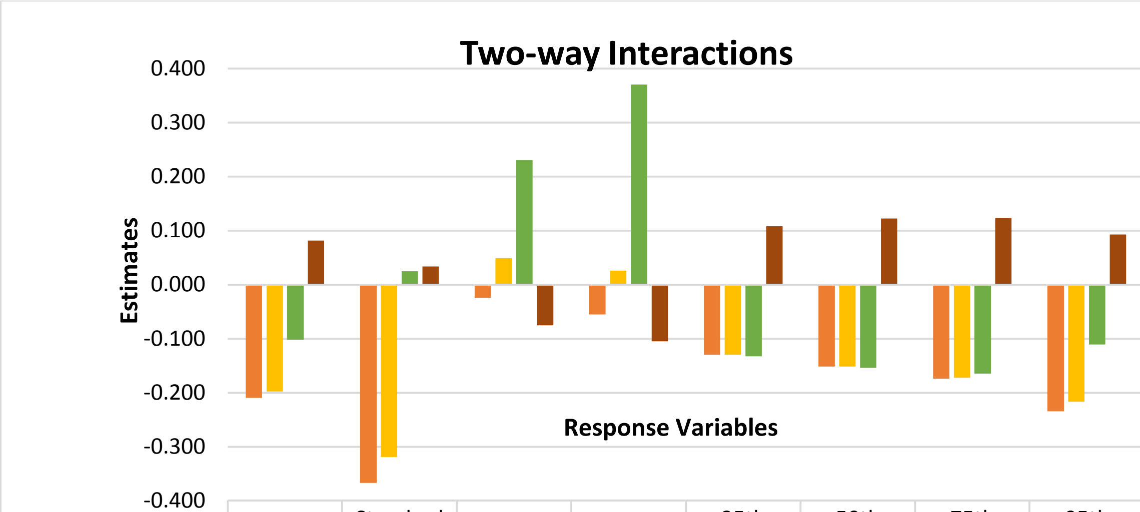


Figure 23. Main Effects plot of dispatching policies on skewness of cycle time distribution for the mini-fab model.

1.2 *Two-way interactions for mini-fab model common across all response variables:*

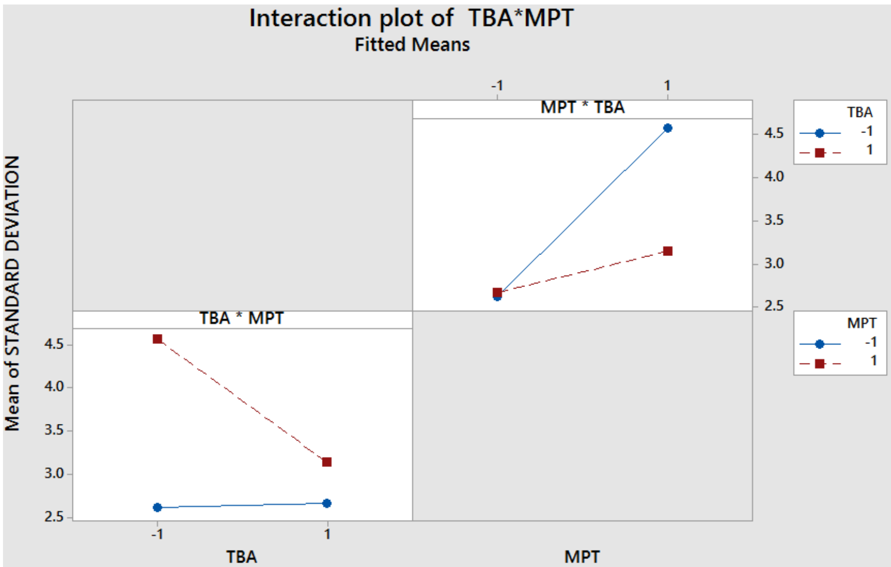
The two-way interactions help understand how the level of one variable impacts the influence of the other variable on a response. Figure 24 summarizes the significant two way interactions common across all the response variables. The figure illustrates that the two-way interaction with the largest magnitude is the interaction between MPT and dispatching policies on the kurtosis of the cycle time distribution. The two-way interaction with the smallest magnitude of two-way interaction is the interaction between MPT and dispatching policies on the standard deviation of the cycle time distribution.



	Mean	Standard Deviation	Skewness	Kurtosis	25th Percentile	50th Percentile	75th Percentile	95th Percentile
TBA* MPT	-0.210	-0.367	-0.025	-0.055	-0.130	-0.152	-0.174	-0.234
Batch Size* MPT	-0.198	-0.320	0.049	0.026	-0.130	-0.152	-0.173	-0.217
MPT* DP	-0.102	0.025	0.231	0.370	-0.133	-0.154	-0.165	-0.111
Batch Size* DP	0.082	0.034	-0.075	-0.105	0.108	0.122	0.123	0.093

Figure 24. Interaction effects of TBA, MPT, batch size and dispatching policies on all response variables for the mini-fab model.

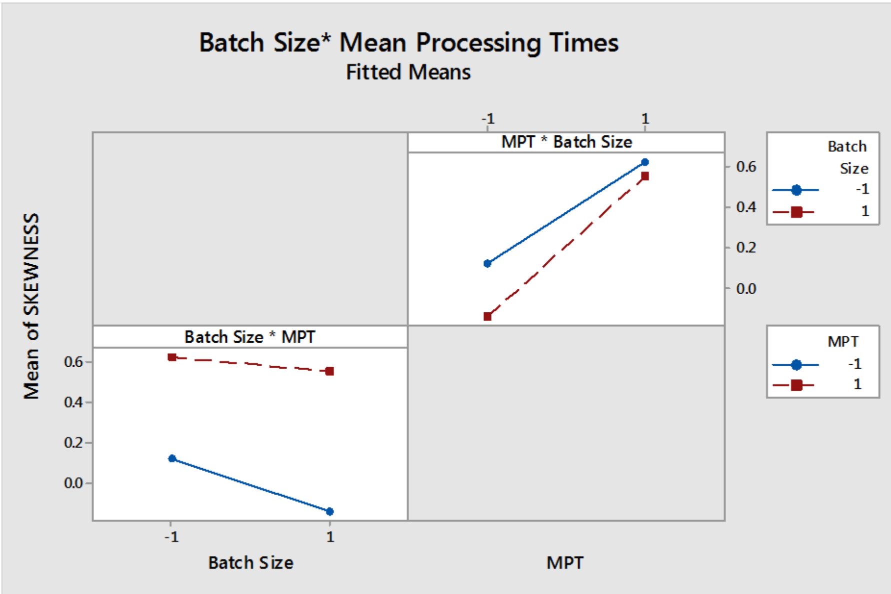
Figure 22 illustrates the effect of the interaction of TBA and MPT on the standard deviation of the cycle time distribution. The interaction demonstrates that the effect of increasing MPT on the standard deviation of the cycle time distribution is more pronounced when the TBA in the system is shorter. A corresponding impact (greater increase when TBA is shorter than when TBA is longer) can be seen for the mean, skewness, kurtosis and percentiles of the cycle time distribution. Based on inferences from factory physics (Hopp and Spearman, 2001), these effects are not surprising. As jobs arrive more frequently to a production system (i.e., when TBA is at a lower value), the overall utilization of the bottleneck within the system increases. Correspondingly, the theory of constraints (Hopp and Spearman, 2001) underscores the idea that reducing the impact of these bottlenecks has a direct increase on the system throughput and that this impact is more pronounced when the bottleneck is more severe, as would be the case



when the MPT increases.

Figure 22. Interaction plot of TBA and MPT on standard deviation of cycle time distribution for mini-fab model.

The interaction between batch size and MPT has the greatest effect on standard deviation and the least effect on kurtosis (effect size for standard deviation is 0.319 and that for kurtosis is 0.026). Figure 23 shows the effect of this interaction on the skewness of cycle time distribution. This interaction demonstrates that the effect on the skewness of the cycle time distribution of changing MPT is more pronounced when the batch size is smaller. This is also true for the mean, standard deviation, kurtosis and percentiles of the cycle time distribution. When the batch size is smaller, the queue in front of the batching tool is longer; correspondingly, the effect of increasing MPT, is to increase the cycle time



of the system (Hopp and Spearman, 2001).

Figure 23. Interaction plot of batch size and MPT on skewness of cycle time of distribution for the mini-fab model.

The effect of interaction between batch size and dispatching polices is greatest on the kurtosis (effect size= 0.370) of the cycle time distribution and is the least on the mean (effect size= 0.102) of the cycle time distribution. Figure 24, Figure 25 and Figure 26 visually display the impact of this on interaction on standard deviation, skewness and 25th percentile of cycle time distribution respectively. The effect of changing the dispatching policies on the standard deviation of the cycle time distribution is more pronounced when the batch size is smaller. This is also the case for mean, skewness, kurtosis and percentiles of the cycle time distribution. Figures 27 to 29 also illustrate that the slope of the line connecting the two types of dispatching policies when batch size is smaller is smaller for the standard deviation of the cycle time distribution than skewness and the 25th percentile. Furthermore, the slopes for skewness and 25th percentile are opposite in direction. When the batch size is smaller, more jobs are waiting to get batched because the machine is processing batches more frequently.

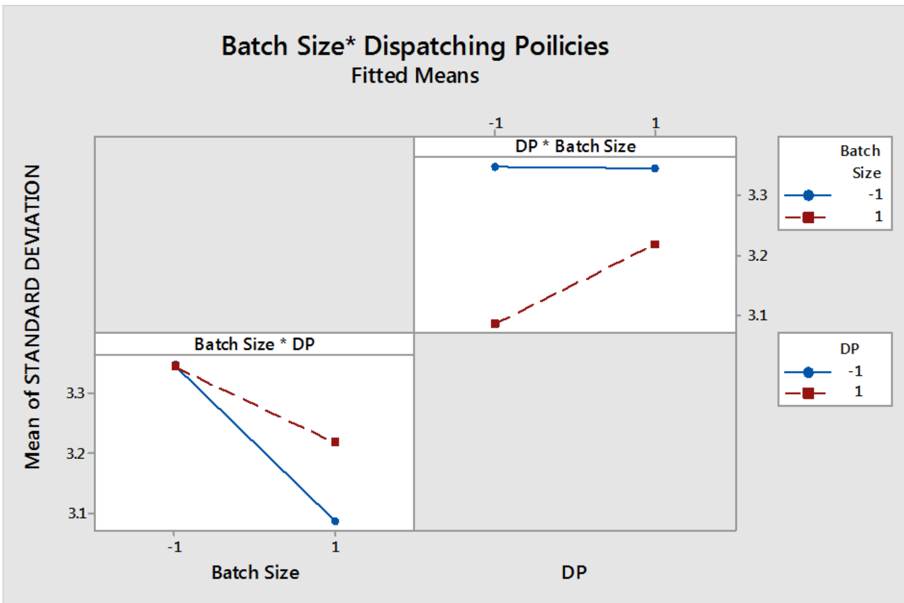


Figure 24. Interaction plot of batch size and dispatching policies on standard deviation of cycle time of distribution for the mini-fab model.

The interaction between MPT and dispatching policies has the greatest impact on the kurtosis of the cycle time distribution and the smallest impact on the standard deviation of the cycle time distribution. Figure 30 illustrates the interaction of MPT and dispatching policies on mean of the cycle time distribution. The effect of changing the dispatching policies on all the response variables is more pronounced when the MPT is at a higher value. One explanation for this is that the queues are longer when the jobs take longer to get processed (i.e., when MPT is at a higher value). In such case, the effect of changing the dispatching policies is more significant.

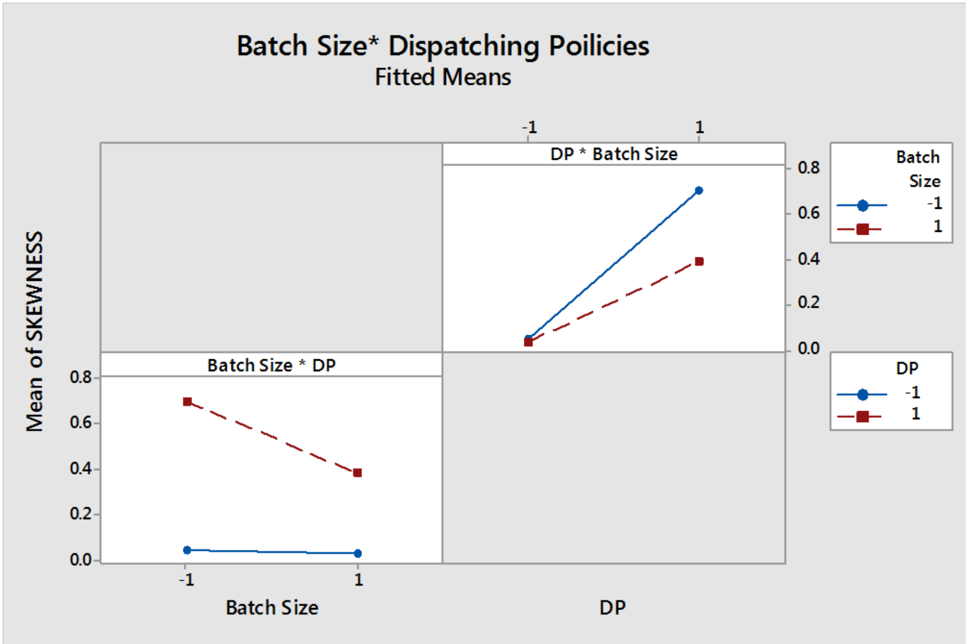


Figure 25. Interaction plot of batch size and dispatching policies on skewness of cycle time of distribution for the mini-fab model.

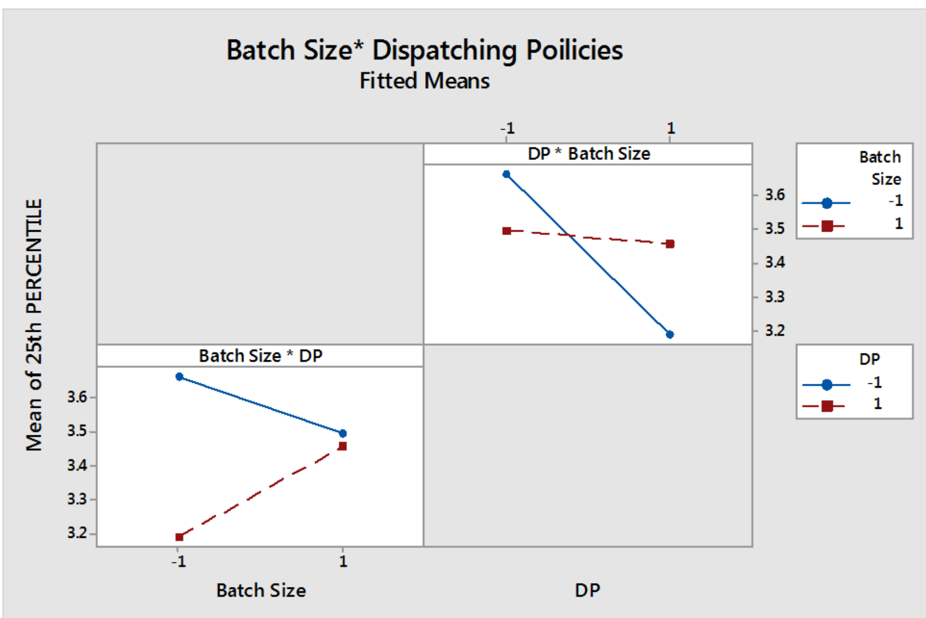


Figure 26. Interaction plot of batch size and dispatching policies on 25th percentile of cycle time of distribution for the mini-fab model.

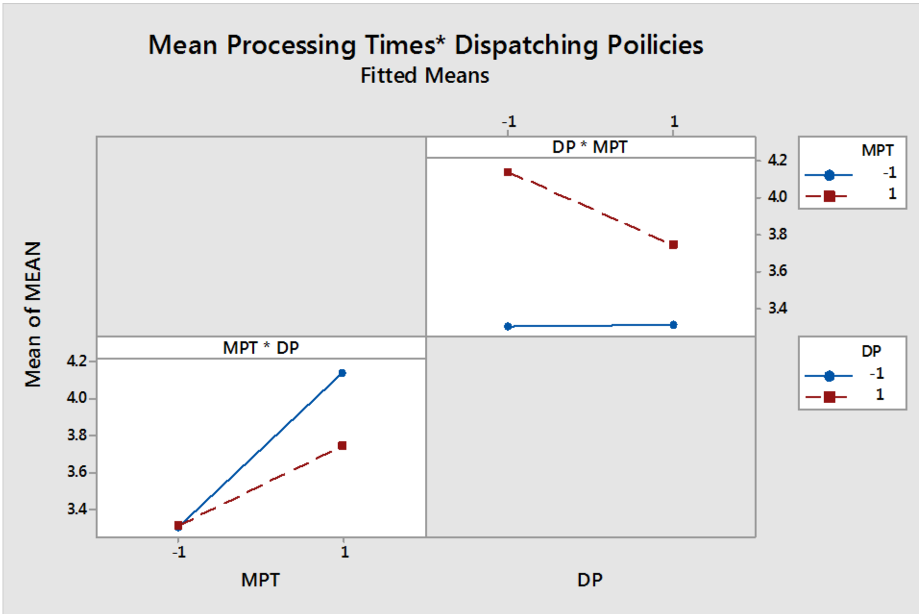
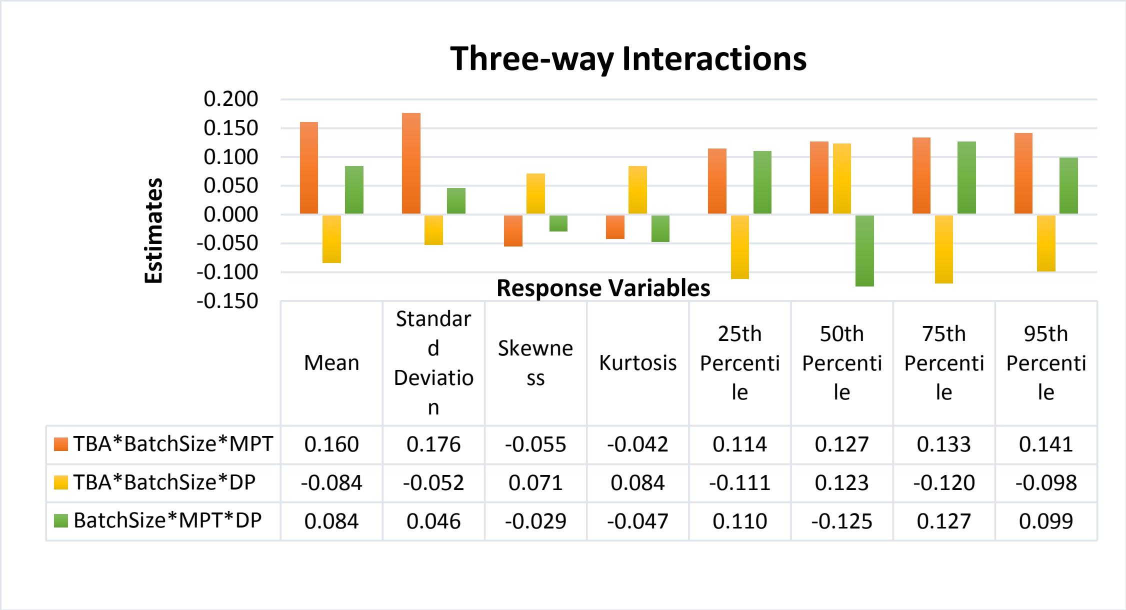


Figure 30. Interaction plot of MPT and dispatching policies on mean of cycle time distribution for the mini-fab model.

After analyzing the significant two-way interactions, the significant three-way interactions that are common across all the response variables were analyzed. These are described next.

1.3. *Three- way interactions for mini-fab model common across all response variables:* There were three three-way interactions that were found to be significant across all response variables. 1) TBA, batch size and MPT, 2) TBA, batch size and dispatching policies, and 3) batch size, MPT and dispatching policies. Figure 31 shows the summary of these interactions. The interaction between TBA, batch size and MPT affect the mean of cycle time distribution the most and kurtosis of the cycle time distribution the least. Secondly, the three-way interaction between TBA, batch size and dispatching policies has a greater effect the percentiles of the cycle time distribution and a weaker effect on the standard deviation of the cycle time distribution. Finally, the three-way interaction between batch size, MPT and dispatching policies has a strong influence on percentiles of the cycle time distribution and a weaker influence on skewness of the cycle time distribution.

An approach suggested by Green and Salkind (2014) in examining three- way interactions was adopted. The analyses suggest examining simple two- way interactions first and then wherever relevant, examining simple main effects. Simple two-way interactions are the significant two- way interaction that differ across various levels of a factor. Simple main effect is the effect of one variable of the significant simple two-way interaction across the levels of another variable. Three-way analyses are done in an effort to understand where the effect of the factors within the interaction produce responses that differ significantly from zero. For every level of the factors involved in a three-way interaction, a significant two-way interaction was identified. For that significant interaction, test of simple main effects was conducted. Simple main effects evaluate the effect of levels of one factor for each level of another factor. The significance level was considered as $\alpha \leq 0.025$ for examining the simple main effects.



	Mean	Standard Deviation	Skewness	Kurtosis	25th Percentile	50th Percentile	75th Percentile	95th Percentile
TBA*BatchSize*MPT	0.160	0.176	-0.055	-0.042	0.114	0.127	0.133	0.141
TBA*BatchSize*DP	-0.084	-0.052	0.071	0.084	-0.111	0.123	-0.120	-0.098
BatchSize*MPT*DP	0.084	0.046	-0.029	-0.047	0.110	-0.125	0.127	0.099

Figure 31. Three- way interaction effects on all response variables for mini-fab model.

First, the simple two-way interaction between batch size and MPT at the two levels of TBA was examined. The interaction was found to be significant at both levels of TBA for mean, standard deviation and the percentiles of the cycle time distribution and non-significant at both levels of TBA for skewness and kurtosis of the cycle time distribution. The p-values for the interaction between batch size and MPT at both levels of TBA are shown in Table 12. Second, the two-way interaction between TBA and MPT at the two levels of batch size were tested. This interaction was also found to be significant at both levels of batch size for mean, standard deviation and the percentiles of the cycle time distribution and non-significant at both levels of batch size for skewness and kurtosis of the cycle time distribution. The p- values for interaction between TBA and MPT at the two levels of batch size are shown in

Table13. Lastly, the simple two-way interaction between TBA and batch size were examined at the two levels of MPT. The interaction was found to be significant at the higher value of MPT and not significant at the lower value of MPT for the mean and percentiles of the cycle time distribution. In contrast, for skewness and kurtosis, the interaction was found to be significant for the lower value of MPT and non-significant for the higher value of MPT. However, for both levels of MPT the interaction is significant for both levels of MPT. The p-values for the interaction between TBA and batch size at different levels of MPT are listed in Table14.

Table 12. Simple two- way interaction between batch size and MPT at levels of TBA for all response variables for the mini-fab model.

Response Variables	**Levels of TBA**	**p- value**
Mean	EXPO (225) mins EXPO (260) mins	0.000 0.000
Standard Deviation	EXPO (225) mins EXPO (260) mins	0.000 0.000
Skewness	EXPO (225) mins EXPO (260) mins	0.199 0.846
Kurtosis	EXPO (225) mins EXPO (260) mins	0.490 0.759
25th Percentile	EXPO (225) mins EXPO (260) mins	0.001 0.000
50th Percentile	EXPO (225) mins EXPO (260) mins	0.000 0.000
75th Percentile	EXPO (225) mins EXPO (260) mins	0.000 0.000
95th Percentile	EXPO (225) mins EXPO (260) mins	0.000 0.000

Table 13. Simple two- way interaction between TBA and MPT at levels of batch size for all response variables for the mini-fab model.

Response Variables	Levels of batch size	p- value
Mean	3 10	0.000 0.000
Standard Deviation	3 10	0.000 0.000
Skewness	3 10	0.882 0.027
Kurtosis	3 10	0.886 0.107
25th Percentile	3 10	0.001 0.001
50th Percentile	3 10	0.000 0.000
75th Percentile	3 10	0.000 0.000
95th Percentile	3 10	0.000 0.000

Table 14. Simple two- way interaction between TBA and batch size at levels of MPT for all response variables for the mini-fab model.

Response Variables	Levels of MPT	p- value
Mean	-30% of mean +30% of mean	0.596 0.000
Standard Deviation	-30% of mean +30% of mean	0.090 0.000
Skewness	-30% of mean +30% of mean	0.003 0.391

Kurtosis	-30% of mean +30% of mean	0.004 0.791
25th Percentile	-30% of mean +30% of mean	0.730 0.010
50th Percentile	-30% of mean +30% of mean	0.500 0.010
75th Percentile	-30% of mean +30% of mean	0.376 0.010
95th Percentile	-30% of mean +30% of mean	0.257 0.000

Figure 32 illustrates the significant simple two-way interaction between TBA and batch size for the mean of the cycle time distribution when MPT is held at its higher value. Figure 33 shows the same interaction for skewness when MPT is held at its lower value. Although these interactions are significant at different values of MPT, the impact of change in the batch size is more pronounced when TBA is at a lower value.

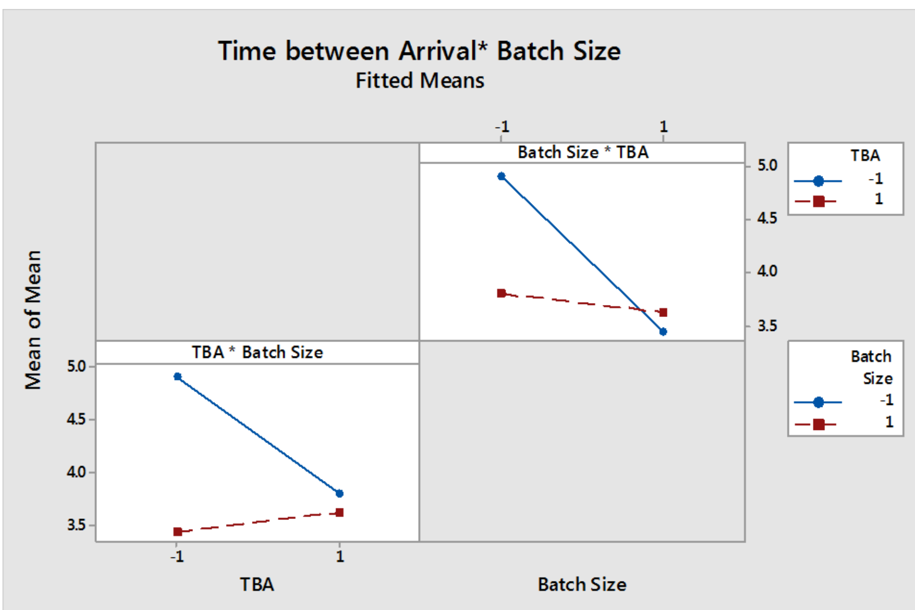


Figure 32. Interaction plot between TBA and batch size on the mean of the cycle time distribution when MPT is held at its higher value for the mini-fab model.

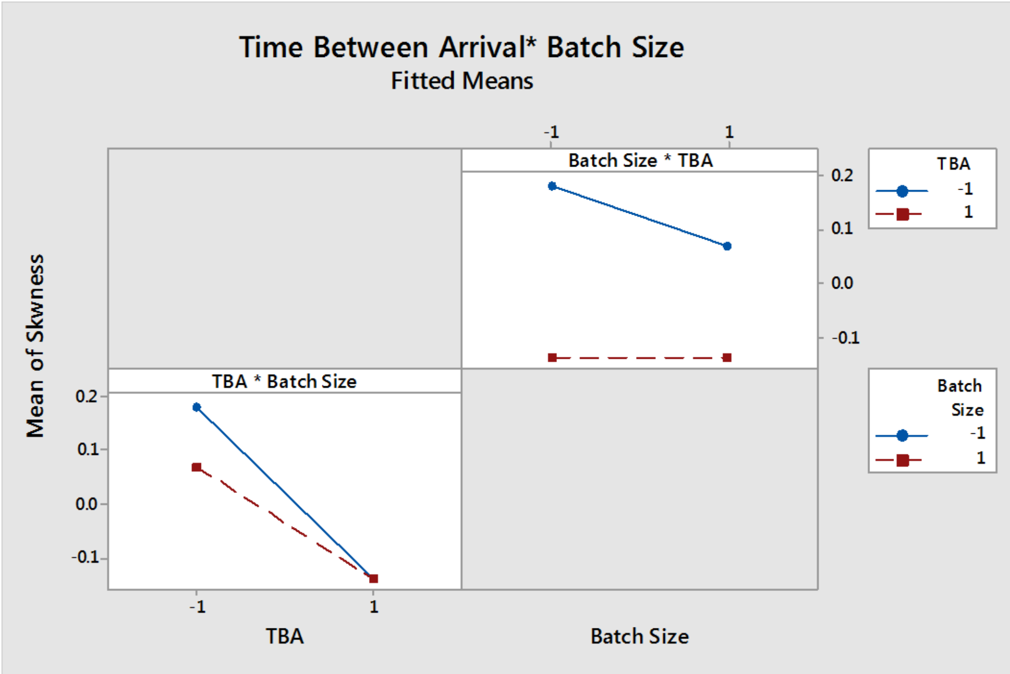


Figure 33. Interaction plot between TBA and batch size on the skewness of the cycle time distribution when MPT is held at its lower value for the mini-fab model.

Once the significant simple two-way interactions were found, further analyses of simple main effects were conducted to investigate the significant interaction between TBA and batch size. The results for this analyses are shown in Table 15. At every level of MPT (for which the simple two-way interaction is significant), Table 15 shows the p-values for the simple main effects of batch size at different levels of TBA. Also, further Table 15 shows the p-values for the simple main effects of TBA at different levels of batch size. The results show that when MPT and TBA are at higher value, the simple main effect of changing batch size for mean and percentiles of cycle time distribution is not statistically significant and when TBA is at a lower value the simple main effect of change in batch size is significant. When MPT is at higher value, the main effect of change in TBA is significant for both levels of batch size. When MPT is at a lower value, the main effect of change in batch size for skewness and kurtosis of cycle time distribution is not significant at both levels of TBA and main effect of TBA is not significant at higher value of batch size. The main effect of change in TBA is however significant when MPT and batch size are at lower values for skewness and kurtosis of cycle time distribution. This is illustrated in Figure 32 and Figure 33.

Table 15. Simple main effects analyses for the interaction between TBA and batch size for three- way interaction between TBA, batch size and MPT for the mini-fab model.

MPT value: +30% of mean (Higher value)			
Simple main effects for batch size			
TBA: EXPO (260) mins (Higher value)		**TBA: EXPO (225) mins (Lower value)**	
Response Variables	**p- value**	**Response Variables**	**p- value**
Mean	0.272	Mean	0.006
25th percentile	0.374	25th percentile	0.000
50th percentile	0.209	50th percentile	0.000
75th percentile	0.048	75th percentile	0.000
95th percentile	0.071	95th percentile	0.002
Simple main effects for TBA			
Batch size: 10 (Higher value)		**Batch size:3 (Lower value)**	
Response Variables	**p- value**	**Response Variables**	**p- value**
Mean	0.000	Mean	0.000
25th percentile	0.021	25th percentile	0.001
50th percentile	0.006	50th percentile	0.001
75th percentile	0.001	75th percentile	0.000
95th percentile	0.000	95th percentile	0.000
MPT value: -30% of mean (Lower value)			
Simple main effects for batch size			
TBA: EXPO (260) mins (Higher value)		**TBA: EXPO (225) mins (Lower value)**	
Response Variables	**p- value**	**Response Variables**	**p- value**
Skewness	0.907	Skewness	0.171
Kurtosis	0.785	Kurtosis	0.027
Simple main effects for TBA			
Batch size: 10 (Higher value)		**Batch size:3 (Lower value)**	
Response Variables	**p- value**	**Response Variables**	**p- value**
Skewness	0.952	Skewness	0.003

Kurtosis	0.171	Kurtosis	0.004

The same approach for analyzing the three-way interaction is taken to analyze the interaction between TBA, batch size and dispatching policies and the three-way interaction between batch size, MPT and dispatching policies. For the three-way interaction between TBA, batch size and dispatching policies, the simple two-way interaction between TBA and batch size were found to be significant. The p-values to find the significant simple two-way interaction are shown in Table 16, 17 and 18. The simple two-way interaction between TBA and batch size significant for the mean when the dispatching policies are held at SPT is illustrated in Figure 34. The same interaction significant for the skewness when the dispatching policies are held at FIFO is shown in Figure 35. Further, simple main effects for the interaction between TBA and batch size were examined. The p-values are shown in Table 19.

Now, for the three-way interaction between batch size, MPT and dispatching policies, the simple two-way interaction between batch size and MPT; and the simple two-way interaction between batch size and dispatching policies were found to be significant. The p-values to find the significant two-way interactions for the interactions between batch size, MPT and dispatching policies are listed in Table 20, 21 and 22. The p-values for the analyzes of simple main effects for these interactions are shown in Table 23 and Table 24. Figure 36 illustrates the interaction between batch size and dispatching policies on the skewness of the cycle time distribution when MPT is held at its lower value. The figure demonstrates that the impact of change in dispatching policies is more pronounced when batch size is smaller.

Table 16. Simple two- way interaction between batch size and dispatching policies at levels of TBA for all response variables for the mini-fab model.

Response Variables	Levels of TBA	p- value
Mean	EXPO (225) mins EXPO (260) mins	0.000 0.000
Standard Deviation	EXPO (225) mins EXPO (260) mins	0.000 0.000
Skewness	EXPO (225) mins EXPO (260) mins	0.199 0.846
Kurtosis	EXPO (225) mins EXPO (260) mins	0.490 0.759
25th Percentile	EXPO (225) mins EXPO (260) mins	0.001 0.000
50th Percentile	EXPO (225) mins EXPO (260) mins	0.000 0.000
75th Percentile	EXPO (225) mins EXPO (260) mins	0.000 0.000
95th Percentile	EXPO (225) mins EXPO (260) mins	0.000 0.000

Table 17. Simple two- way interaction between TBA and dispatching policies at levels of batch size for all response variables for the mini-fab model.

Response Variables	**Levels of batch size**	**p- value**
Mean	3 10	0.000 0.000
Standard Deviation	3 10	0.000 0.000
Skewness	3 10	0.882 0.027
Kurtosis	3 10	0.886 0.107
25th Percentile	3 10	0.001 0.001
50th Percentile	3 10	0.000 0.000
75th Percentile	3 10	0.000 0.000
95th Percentile	3 10	0.000 0.000

Table 18. Simple two- way interaction between TBA and batch size at levels of dispatching policies for all response variables for the mini-fab model.

Response Variables	Levels of dispatching policies	p- value
Mean	FIFO SPT	0.06 0.039
Standard Deviation	FIFO SPT	0.032 0.202
Skewness	FIFO SPT	0.003 0.530
Kurtosis	FIFO SPT	0.004 0.762
25th Percentile	FIFO SPT	0.004 0.762
50th Percentile	FIFO SPT	0.004 0.633
75th Percentile	FIFO SPT	0.006 0.321
95th Percentile	FIFO SPT	0.013 0.278

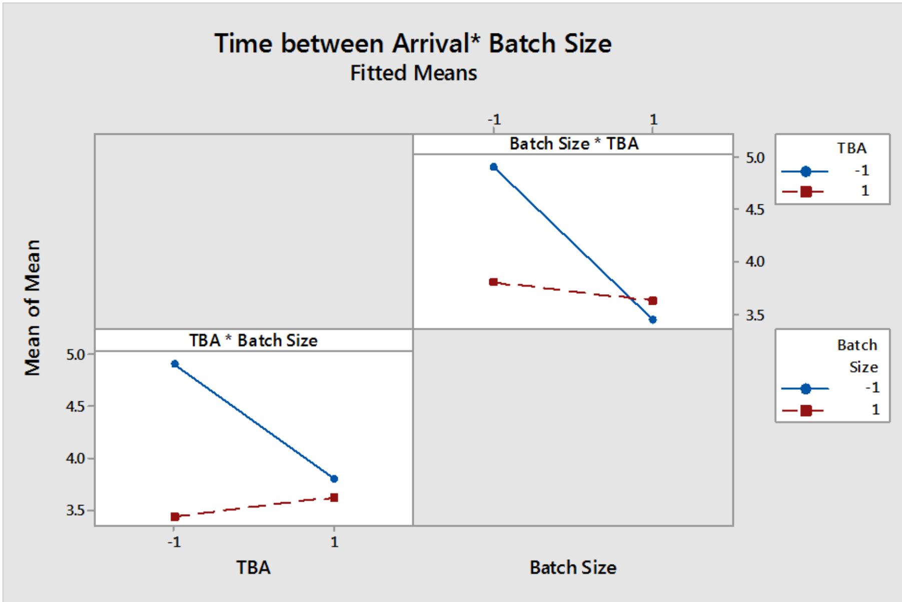


Figure 34. Interaction plot between TBA and batch size on mean of cycle time distribution when dispatching policies are held at SPT for the mini-fab model.

Table 19. Simple main effects analyses for the interaction between TBA and batch size for three- way interaction between TBA, batch size and dispatching policies for the mini-fab model.

Dispatching policy: SPT (Higher value)			
Simple main effects for batch size			
TBA: EXPO (260) mins (Higher value)		**TBA: EXPO (225) mins (Lower value)**	
Response Variables	**p- value**	**Response Variables**	**p- value**
Mean	0.001	Mean	0.000
Simple main effects for TBA			
Batch size: 10 (Higher value)		**Batch size:3 (Lower value)**	
Response Variables	**p- value**	**Response Variables**	**p- value**
Mean	0.186	Mean	0.015
Dispatching policy: FIFO (Lower value)			
Simple main effects for batch size			
TBA: EXPO (260) mins (Higher value)		**TBA: EXPO (225) mins (Lower value)**	
Response Variables	**p- value**	**Response Variables**	**p- value**
Standard Deviation	0.000	Standard Deviation	0.000
Skewness	0.000	Skewness	0.357
Kurtosis	0.000	Kurtosis	0.279
25^{th} Percentile	0.000	25^{th} Percentile	0.000

50^{th} Percentile	0.000	50^{th} Percentile	0.000
75^{th} Percentile	0.000	75^{th} Percentile	0.000
95^{th} Percentile	0.000	95^{th} Percentile	0.000
Simple main effects for TBA			
Batch size: 10 (Higher value)		**Batch size:3 (Lower value)**	
Response Variables	**p- value**	**Response Variables**	**p- value**
Standard Deviation	0.015	Standard Deviation	0.008
Skewness	0.039	Skewness	0.029
Kurtosis	0.070	Kurtosis	0.000
25^{th} Percentile	0.413	25^{th} Percentile	0.004
50^{th} Percentile	0.218	50^{th} Percentile	0.003
75^{th} Percentile	0.084	75^{th} Percentile	0.003
95^{th} Percentile	0.033	95^{th} Percentile	0.005

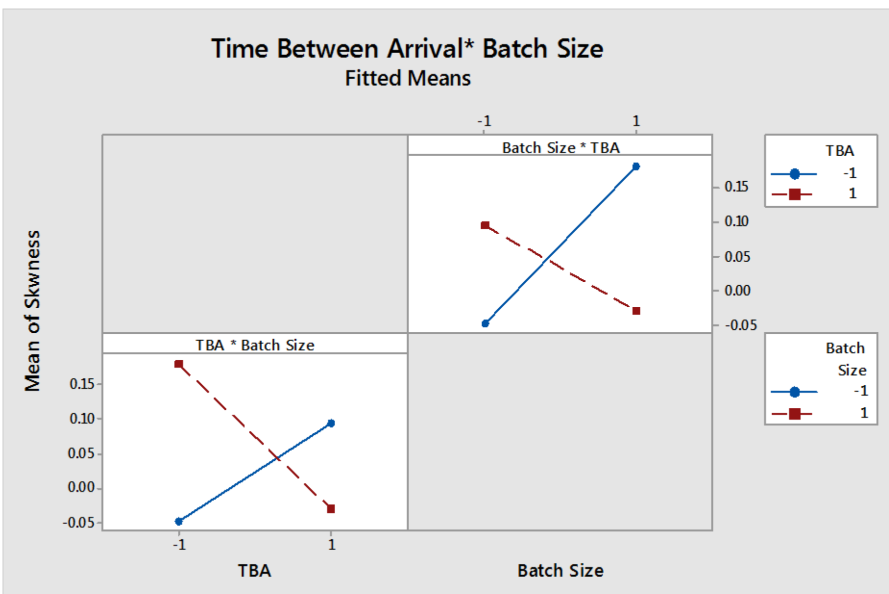


Figure 35. Interaction plot between TBA and batch size on skewness of cycle time distribution when dispatching policies are held at FIFO for the mini-fab model.

Table 20. Simple two- way interaction between MPT and dispatching policies at levels of batch size for all response variables for the mini-fab model.

Response Variables	Levels of batch size	p- value
Mean	3 10	0.000 0.000
Standard Deviation	3 10	0.000 0.000
Skewness	3 10	0.882 0.027
Kurtosis	3 10	0.886 0.107
25th Percentile	3 10	0.001 0.001
50th Percentile	3 10	0.000 0.000
75th Percentile	3 10	0.000 0.000
95th Percentile	3 10	0.000 0.000

Table 21. Simple two- way interaction between batch size and dispatching policies at levels of MPT for all response variables for the mini-fab model.

Response Variables	Levels of MPT	p- value
Mean	-30% of mean +30% of mean	0.596 0.000
Standard Deviation	-30% of mean +30% of mean	0.090 0.000
Skewness	-30% of mean +30% of mean	0.003 0.391
Kurtosis	-30% of mean +30% of mean	0.004 0.791
25th Percentile	-30% of mean +30% of mean	0.730 0.010
50th Percentile	-30% of mean +30% of mean	0.500 0.010
75th Percentile	-30% of mean +30% of mean	0.376 0.010
95th Percentile	-30% of mean +30% of mean	0.257 0.000

Table 22. Simple two- way interaction between batch size and MPT at levels of dispatching policies for all response variables for the mini-fab model.

Response Variables	Levels of dispatching policies	p- value
Mean	FIFO SPT	0.060 0.039
Standard Deviation	FIFO SPT	0.032 0.202
Skewness	FIFO SPT	0.003 0.530
Kurtosis	FIFO SPT	0.000 0.424
25th Percentile	FIFO SPT	0.004 0.762
50th Percentile	FIFO SPT	0.004 0.633
75th Percentile	FIFO SPT	0.006 0.321
95th Percentile	FIFO SPT	0.013 0.278

Table 23. Simple main effects for the interaction between batch size and dispatching policies for three-way interaction between batch size, MPT and dispatching policies for the mini-fab model.

MPT value: +30% of mean (Higher value)			
Simple main effects for dispatching policies			
Batch size: 10 (Higher value)		**Batch size: 3 (Lower value)**	
Response Variables	**p- value**	**Response Variables**	**p- value**
Mean	0.048	Mean	0.026
25^{th} percentile	0.000	25^{th} percentile	0.001
50^{th} percentile	0.000	50^{th} percentile	0.001
75^{th} percentile	0.000	75^{th} percentile	0.001
95^{th} percentile	0.387	95^{th} percentile	0.015
Simple main effects for batch size			
Dispatching policy: SPT (Higher value)		**Dispatching policy: FIFO (Higher value)**	
Response Variables	**p- value**	**Response Variables**	**p- value**
Mean	0.228	Mean	0.012
25^{th} percentile	0.000	25^{th} percentile	0.024
50^{th} percentile	0.000	50^{th} percentile	0.015
75^{th} percentile	0.000	75^{th} percentile	0.011
95^{th} percentile	0.231	95^{th} percentile	0.008
MPT value: -30% of mean (Lower value)			

Simple main effects for dispatching policies			
Batch size: 10 (Higher value)		**Batch size: 3 (Lower value)**	
Response Variables	**p- value**	**Response Variables**	**p- value**
Skewness	0.000	Skewness	0.000
Kurtosis	0.000	Kurtosis	0.000
Simple main effects for batch size			
Dispatching policy: SPT (Higher value)		**Dispatching policy: FIFO (Higher value)**	
Response Variables	**p- value**	**Response Variables**	**p- value**
Skewness	0.000	Skewness	0.000
Kurtosis	0.000	Kurtosis	0.000

Table 24. Simple main effects for the interaction between batch size and MPT for three-way interaction between batch size, MPT and dispatching policies for the mini-fab model.

Dispatching policy: SPT (Higher value)			
Simple main effects for MPT			
Batch size: 10 (Higher value)		**Batch size: 3 (Lower value)**	
Response Variables	**p- value**	**Response Variables**	**p- value**
Mean	0.000	Mean	0.000
Simple main effects for batch size			
MPT: +30% of mean (Higher value)		**MPT: -30% of mean (Lower value)**	
Response Variables	**p- value**	**Response Variables**	**p- value**
Mean	0.228	Mean	0.000
Dispatching policy: FIFO (Lower value)			
Simple main effects for MPT			
Batch size: 10 (Higher value)		**Batch size: 3 (Lower value)**	
Response Variables	**p- value**	**Response Variables**	**p- value**
Standard Deviation	0.000	Standard Deviation	0.000
Skewness	0.000	Skewness	0.801
Kurtosis	0.000	Kurtosis	0.936
25th Percentile	0.000	25th Percentile	0.000
50th Percentile	0.000	50th Percentile	0.000
75th Percentile	0.000	75th Percentile	0.000
95th Percentile	0.000	95th Percentile	0.000
Simple main effects for TBA			
Batch size: 10 (Higher value)		**Batch size:3 (Lower value)**	
Response Variables	**p- value**	**Response Variables**	**p- value**

Standard Deviation	0.018	Standard Deviation	0.000
Skewness	0.003	Skewness	0.000
Kurtosis	0.002	Kurtosis	0.000
25th Percentile	0.000	25th Percentile	0.000
50th Percentile	0.000	50th Percentile	0.000
75th Percentile	0.000	75th Percentile	0.000
95th Percentile	0.231	95th Percentile	0.000

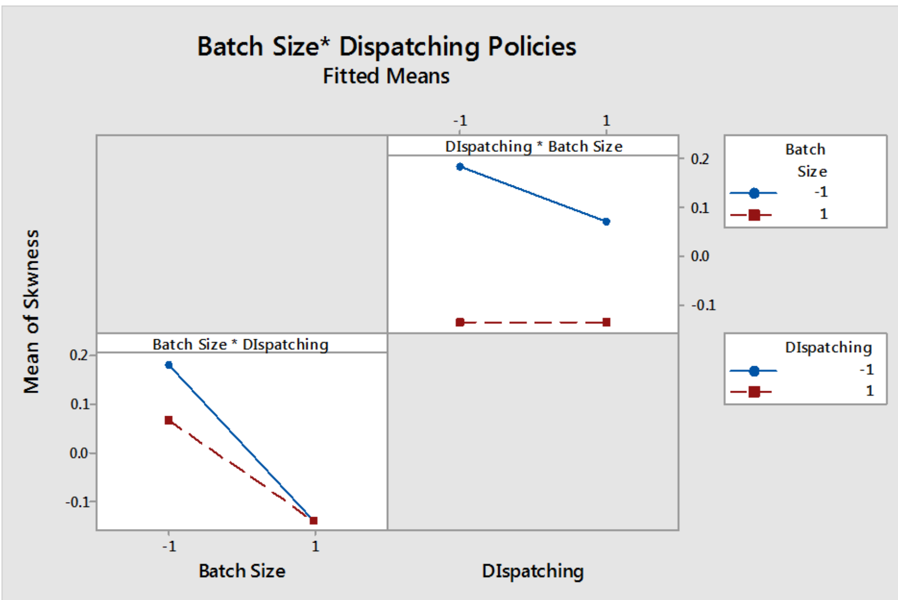


Figure 36. Interaction plot between batch size and dispatching policies on the skewness of the cycle time distribution when MPT is held at its lower value for the mini-fab model.

This completes the analyzes of the three-way interactions for the mini-fab model common across all response variables. There are some factors and interactions that are statistically significant ($p \leq 0.05$) specifically to a response variable for the mini-fab model. Section 1.4 lists and describes the main effects and interactions significant to each of the response variables considered in this research.

1.4 ***Main effects and two-interactions significant only for a subset of the response variables for the mini-fab model:***

For the response variable describing the mean of the cycle time distribution, the main effects found to be significant, apart from those described in section 1.1, are batch size, COV and mean time to repair in EM. The effect of batch size is to decrease the mean of the cycle time distribution and that of COV and mean time to repair in EM is to increase the mean of the cycle time distribution.

Apart from the ones mentioned in section 1.2, the significant two-way interactions for the mean of the cycle time distribution are listed in Table 25. The effect sizes and p-values are also listed in Table .

Table 25. Two-way interactions significant specifically to mean of cycle time distribution for the mini-fab model.

Two-way interactions	Effect sizes	p-values
TBA* batch size	0.1619	0.000
TBA* dispatching policies	0.0928	0.000
TBA* COV	-0.0265	0.028
TBA* repair_EM	-0.0320	0.008
Batch size* repair_EM	-0.0270	0.025
MPT* COV	0.0267	0.027
MPT* repair_EM	0.0321	0.008
Dispatching policies* repair_EM	-0.0255	0.034

Figure 37 to 44 illustrate the interaction between TBA and batch size, TBA and dispatching policies, TBA and COV for unloading and repair time in emergency failures; and; TBA and mean time to repair in emergency failures respectively. All the interactions demonstrate that the effect of changing batch size, dispatching policies, COV for unloading and repair in EM from lower value to higher value on the mean of cycle time distribution is more pronounced when the TBA is at lower value.

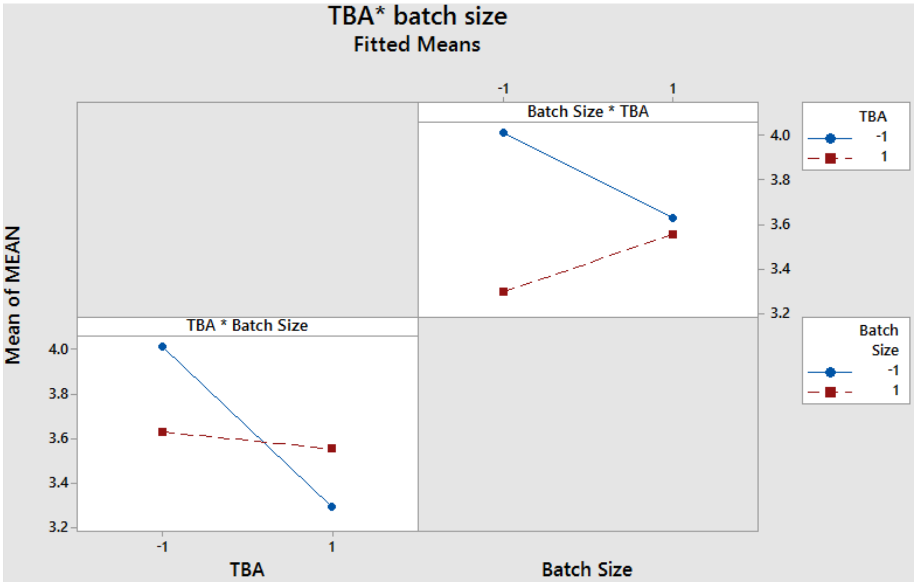


Figure 37. Two-way interaction between TBA and batch size for the mean of the cycle time distribution for the mini-fab model.

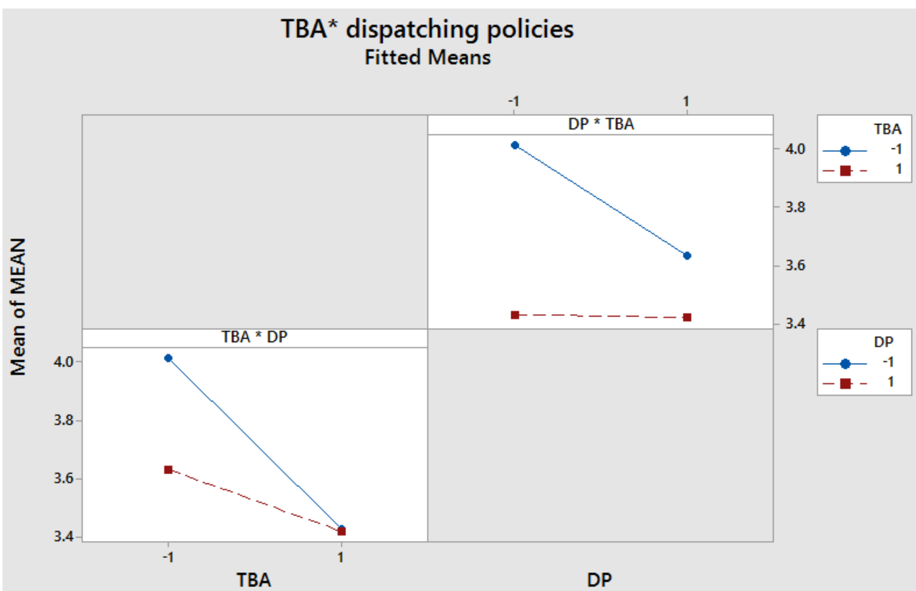


Figure 78. Two-way interaction between TBA and dispatching policies for the mean of the cycle time distribution for the mini-fab model.

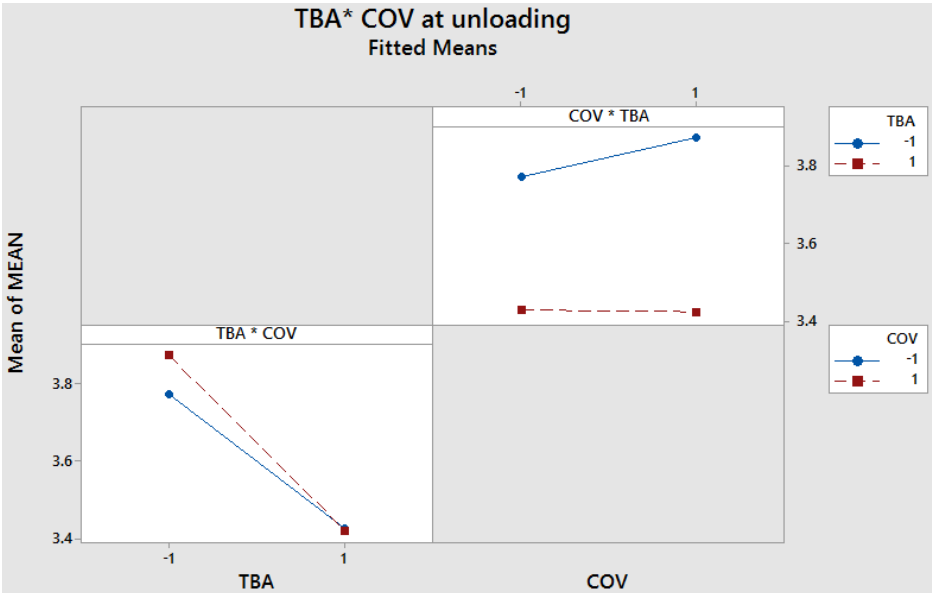


Figure 39. Two-way interaction between TBA and COV for the mean of the cycle time distribution for the mini-fab model.

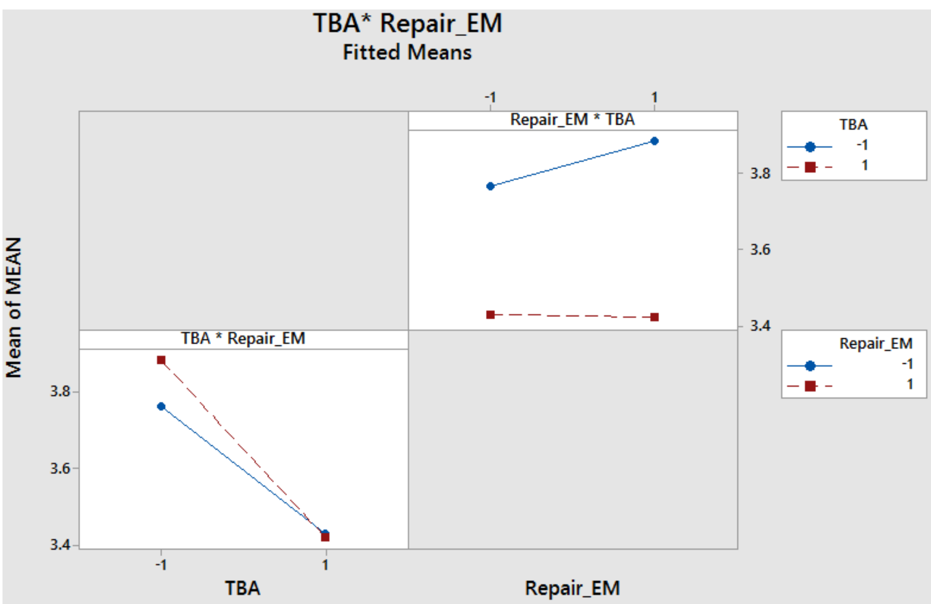
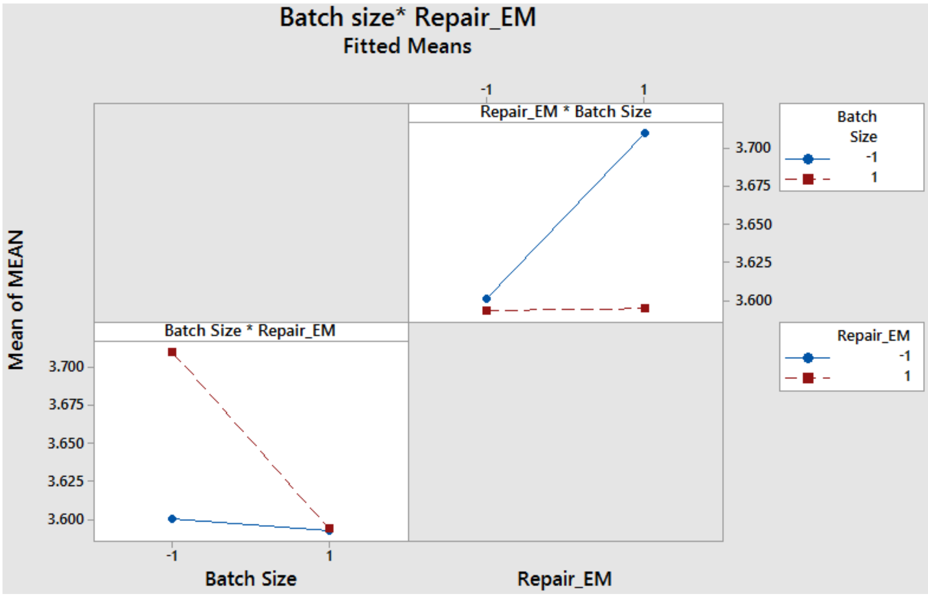


Figure 40. Two-way interaction between TBA and mean time to repair in EM for the mean of the cycle time distribution for the mini-fab model.

Figure 41 demonstrates the interaction between batch size and mean time to repair in EM. The effect of changing the mean of repair time at EM from a lower value of the higher value on the mean of cycle time distribution in greater when the batch size is at



lower value.

Figure 41. Two-way interaction between batch size and mean time to repair in EM for the mean of the cycle time distribution for the mini-fab model.

Figure 42 and 43 illustrate the interaction between MPT and COV for unloading and MPT and mean time to repair in EM respectively. The interactions demonstrate that the effect of changing the COV and mean time to repair in EM from lower value to higher value on the mean of the cycle time distribution is more prominent when MPT is at a higher value.

Figure 44 illustrates the interaction between dispatching policies and mean time to repair in EM showing that the change in the mean time to repair in EM from lower value to higher on the mean of cycle time distribution is greater when the dispatching policies are set to FIFO.

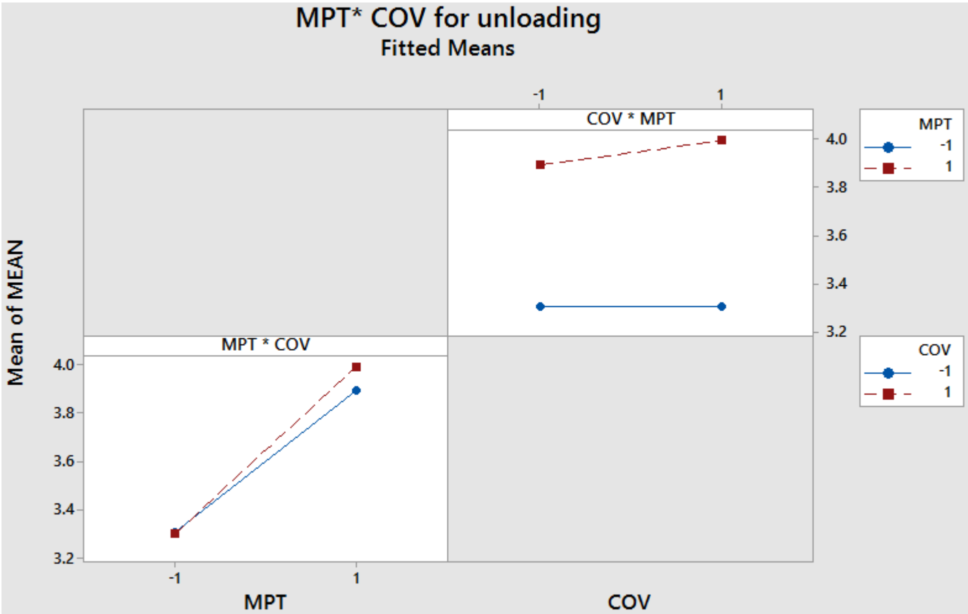


Figure 42. Two-way interaction between MPT and COV for the mean of the cycle time distribution for the mini-fab model.

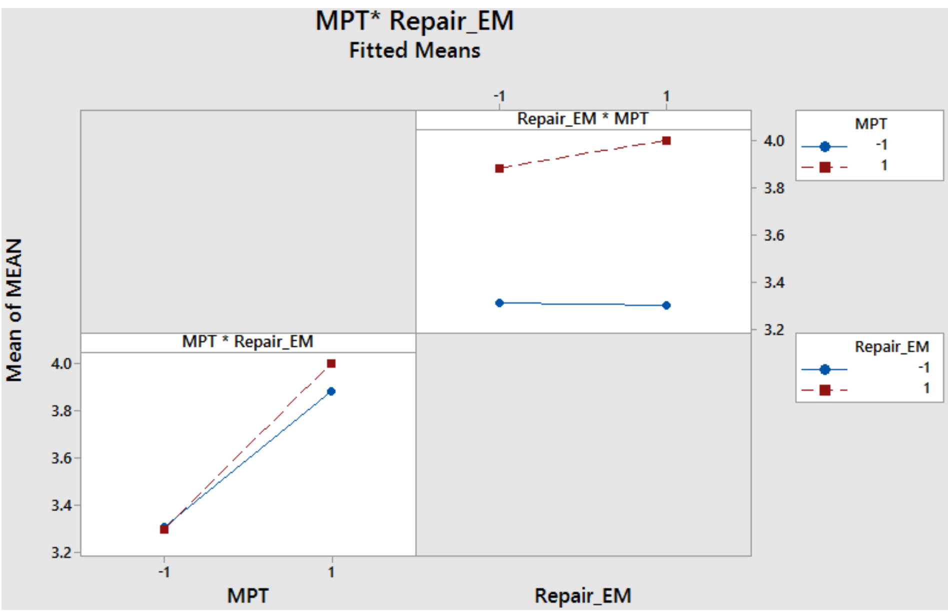


Figure 43. Two-way interaction between MPT and mean time to repair in EM for the mean of the cycle time distribution for the mini-fab model.

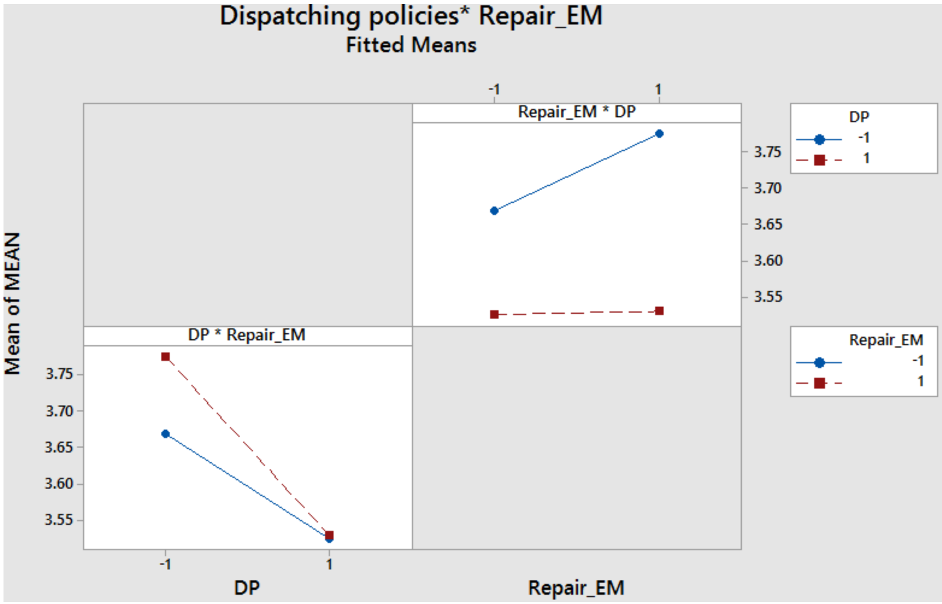


Figure 44. Two-way interaction between dispatching policies and mean time to repair in EM for the mean of the cycle time distribution for the mini-fab model.

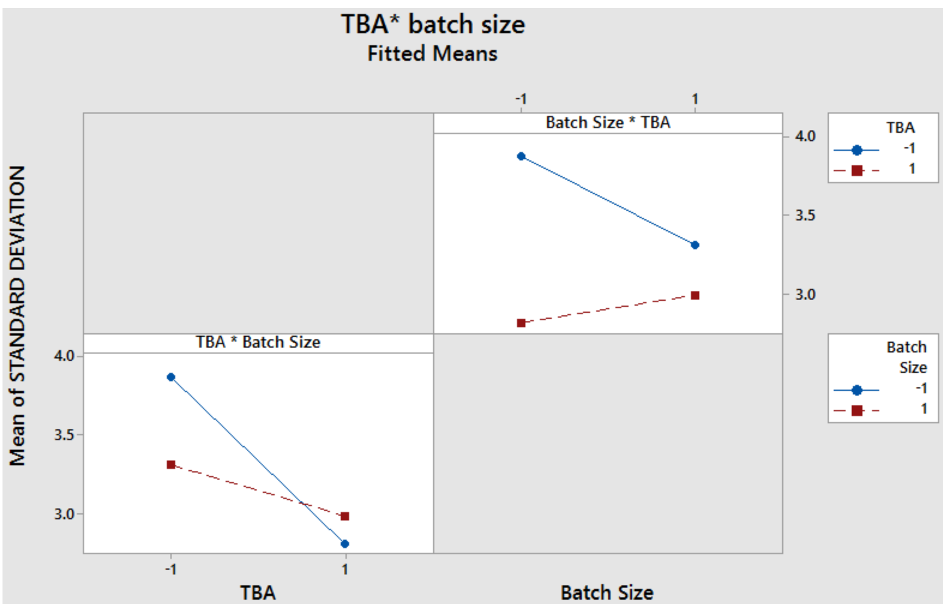
Now, moving forward to standard deviation of the cycle time distribution, the main effects significant specifically to standard deviation are batch size, COV and mean time to repair in EM. The effect of batch size is to decrease the standard deviation of the cycle time distribution and the effect of COV and mean time to repair in EM is to increase the standard deviation of the cycle time distribution. Table 26 shows the two-way interactions that are significant to standard deviation of the cycle time distribution only.

Table 26. Two-way interactions significant specifically to standard deviation of cycle time distribution for the mini-fab model**.**

Two-way interactions	**Effect sizes**	**p-values**
TBA* batch size	0.18363	0.000
TBA* COV	-0.02881	0.001
TBA* repair_EM	-0.03339	0.000
Batch size* COV	-0.01919	0.020

Batch size* repair_EM	-0.02550	0.002
MPT* COV	0.02907	0.001
MPT* repair_EM	0.03172	0.000
Dispatching policies* repair_EM	-0.01971	0.017

Figure ***8***45, Figure ***9***46 and Figure ***10***47 illustrate the interactions between TBA and batch size, TBA and COV and TBA and mean time to repair in EM respectively. The interactions demonstrate that when batch size, COV and mean time to repair in EM change from lower value to higher value, the effect on the standard deviation of cycle



time distribution is greater when TBA is at a lower value.

Figure 8. Two-way interaction between TBA and batch size for the standard deviation of the cycle time distribution for the mini-fab model.

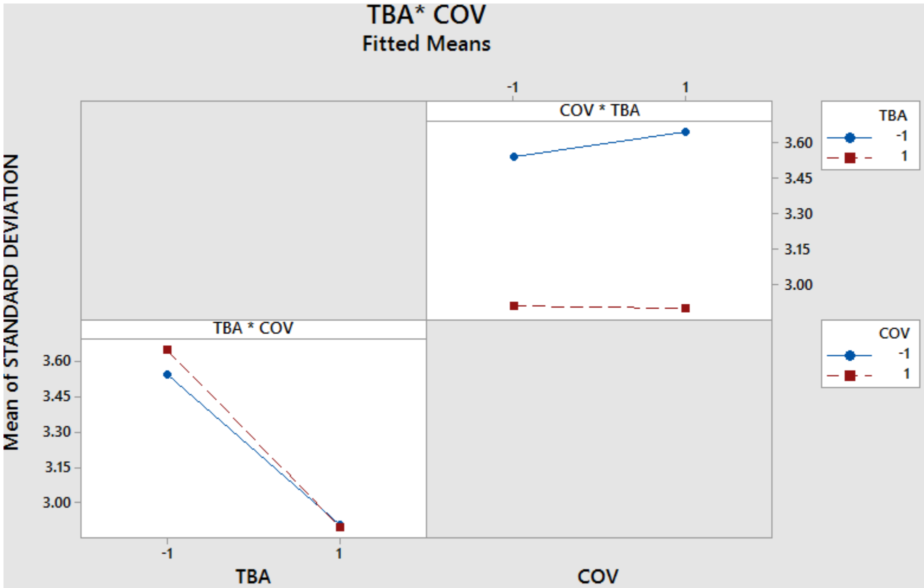


Figure 9. Two-way interaction between TBA and COV for the standard deviation of the cycle time distribution for the mini-fab model.

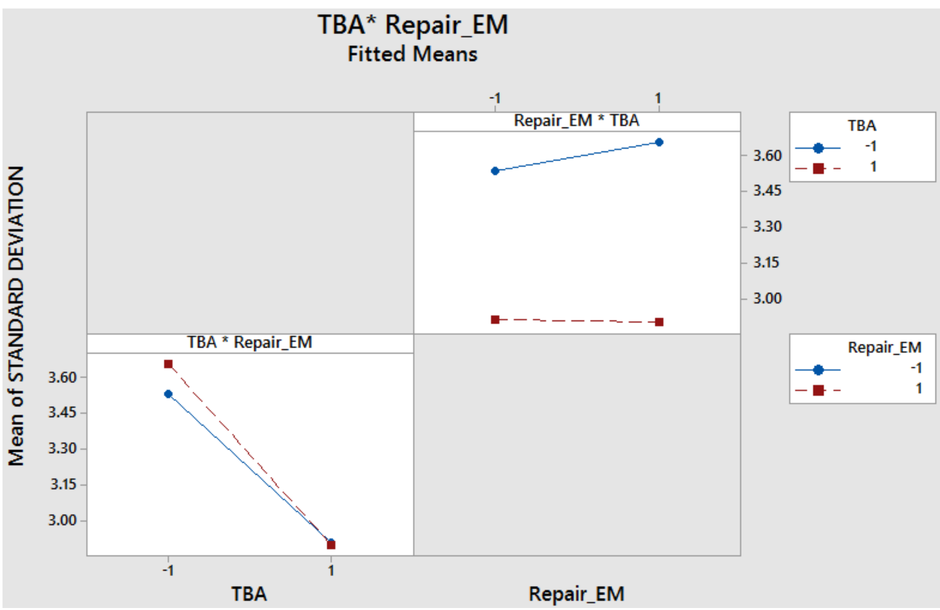


Figure 10. Two-way interaction between TBA and mean time to repair in EM for the standard deviation of the cycle time distribution for the mini-fab model.

Figure ***11***48 and Figure ***12***49 illustrate the interaction between batch size and COV; and batch size and mean time to repair in EM respectively. The interactions show that the effect of change in COV and mean time to repair in EM on the standard deviation of the cycle time distribution is greater when batch size is at a lower value.

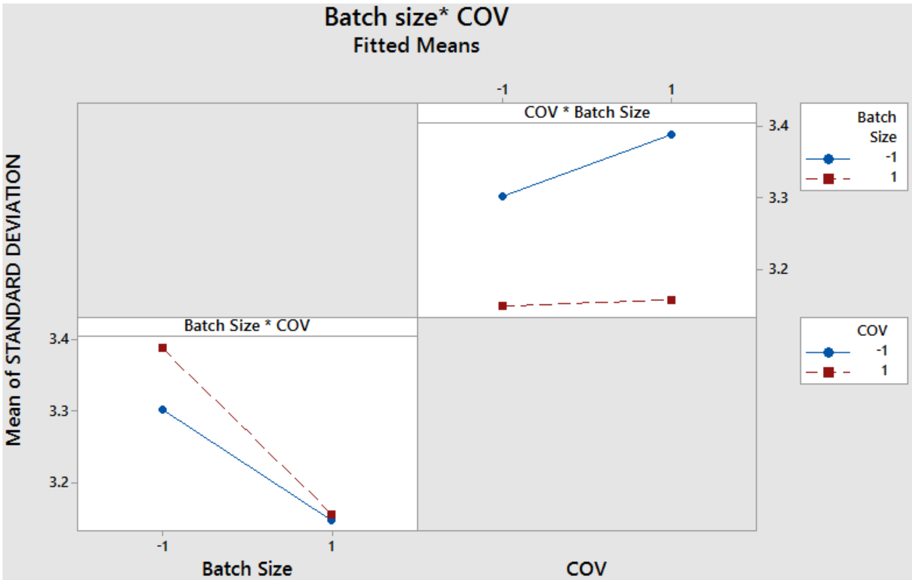


Figure 11. Two-way interaction between batch size and COV for the standard deviation of the cycle time distribution for the mini-fab model.

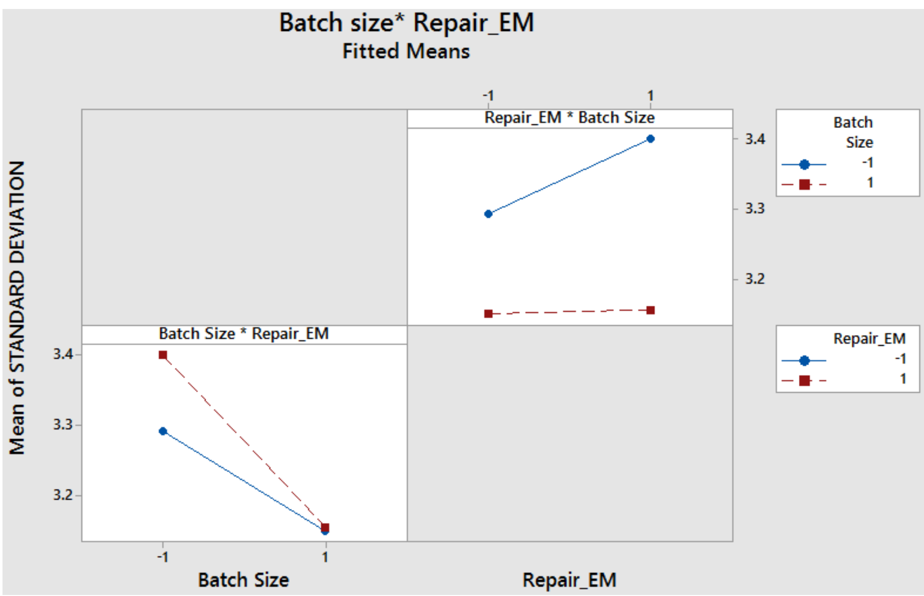


Figure 12. Two-way interaction between batch size and mean time to repair in EM for the standard deviation of the cycle time distribution for the mini-fab model.

For the interactions between MPT and COV; and MPT and Mean time to repair in EM as illustrated in Figure 50 and Figure 51 respectively, it is seen that the effect of changing the COV and mean time to repair in EM on the standard deviation of the cycle time distribution is more when MPT is at a higher value. Figure 52 demonstrates that in the interaction between dispatching policies and mean time to repair in EM, the effect of change in the value of mean time to repair in EM from lower value to higher value on standard deviation of cycle time distribution is more pronounced when the dispatching policies are set to SPT.

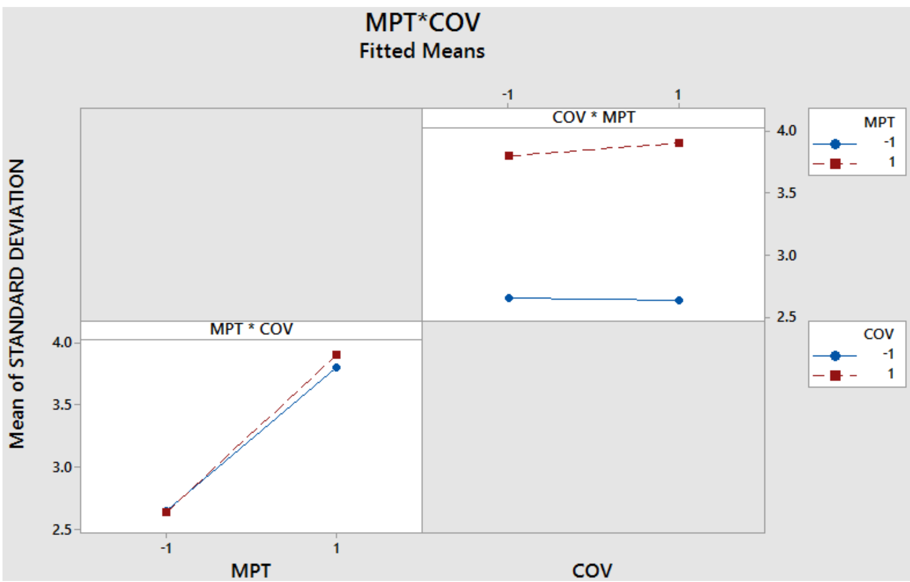


Figure 50. Two-way interaction between MPT and COV for the standard deviation of the cycle time distribution for the mini-fab model.

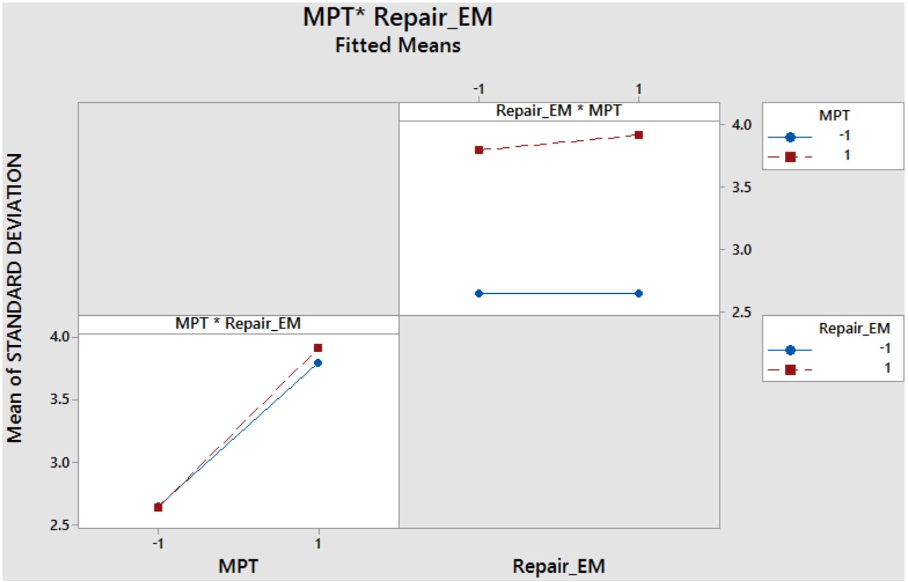


Figure 51. Two-way interaction between MPT and mean time to repair in EM for the standard deviation of the cycle time distribution for the mini-fab model.

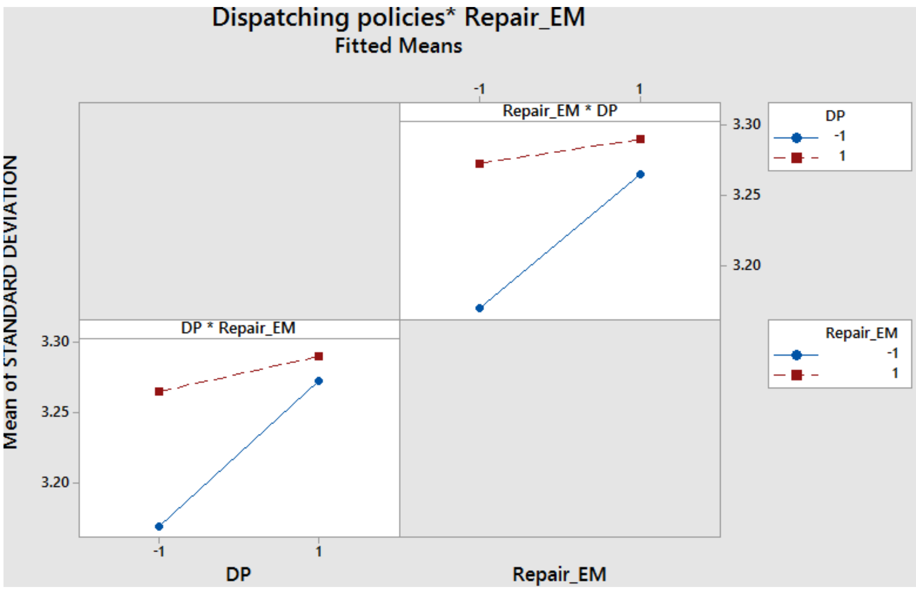


Figure 52. Two-way interaction between dispatching policies and mean time to repair in EM for the standard deviation of the cycle time distribution for the mini-fab model.

Further, for the skewness of the cycle time distribution, apart from the main effects explained in section 1.1, only batch size is significant.

Table 27. Two-way interactions significant specifically to skewness of cycle time distribution for the mini-fab model.

Two-way interactions	**Effect sizes**	**p-values**
TBA* batch size	-0.0270	0.015
TBA* dispatching policies	-0.0880	0.000
MPT* repair_EM	-0.0248	0.025
Dispatching policies* COV	0.0243	0.028
Dispatching policies* repair_EM	0.0252	0.023

Figure 53 to Figure 7 illustrate the interactions mentioned in Table 26. Two-way interactions significant specifically to standard deviation of cycle time distribution for the mini-fab model**.** The interaction between TBA and batch size; and TBA and dispatching policies demonstrate that the changes in batch size and dispatching policies from FIFO to SPT affects the skewness of cycle time distribution more when batch size is at a lower value. The interaction between MPT and mean time to repair in EM shows that the effect of change in mean time to repair in EM is greater on skewness of cycle time distribution when MPT are at a higher value. Lastly, the interactions between dispatching polices and COV; and dispatching polices and mean time to repair in EM indicate that the effect of changes in COV and mean time to repair in EM are greater when dispatching policies are SPT.

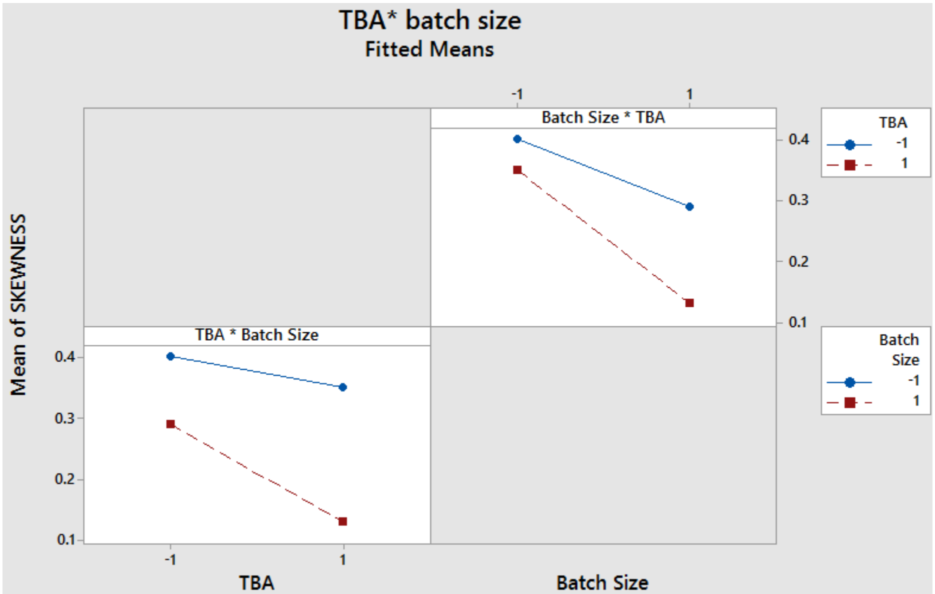


Figure 53. Two-way interaction between TBA and batch size for the skewness of the cycle time distribution for the mini-fab model.

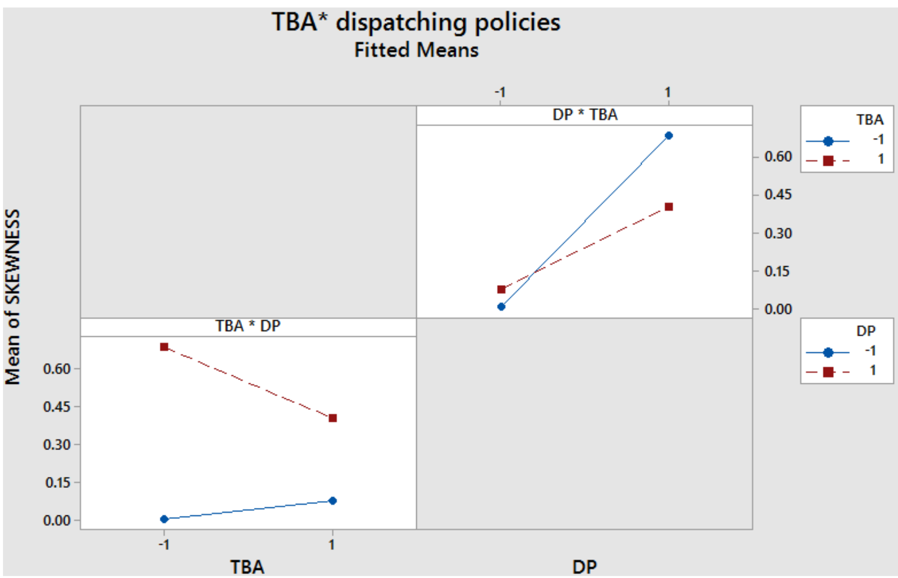


Figure 54. Two-way interaction between TBA and dispatching policies for the skewness of the cycle time distribution for the mini-fab model.

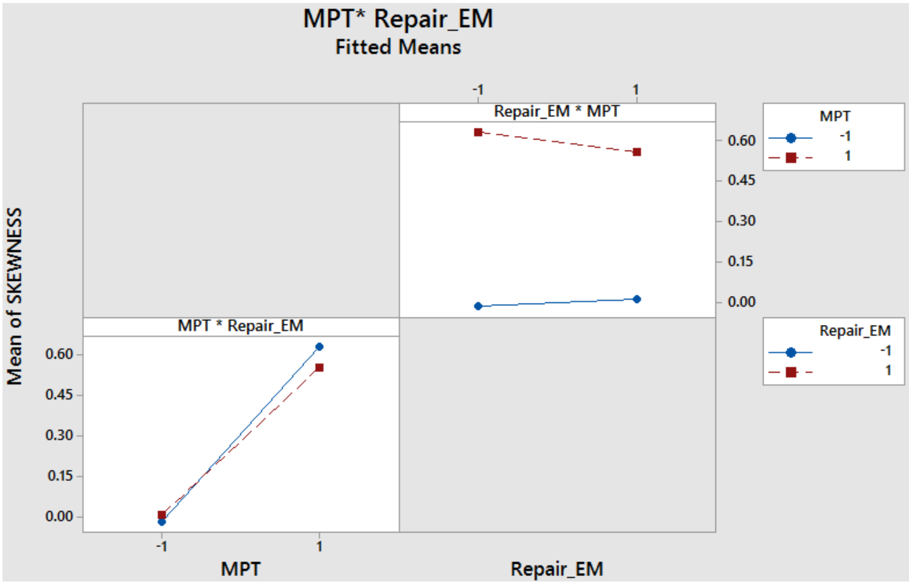


Figure 55. Two-way interaction between MPT and mean time to repair in EM for the skewness of the cycle time distribution for the mini-fab model.

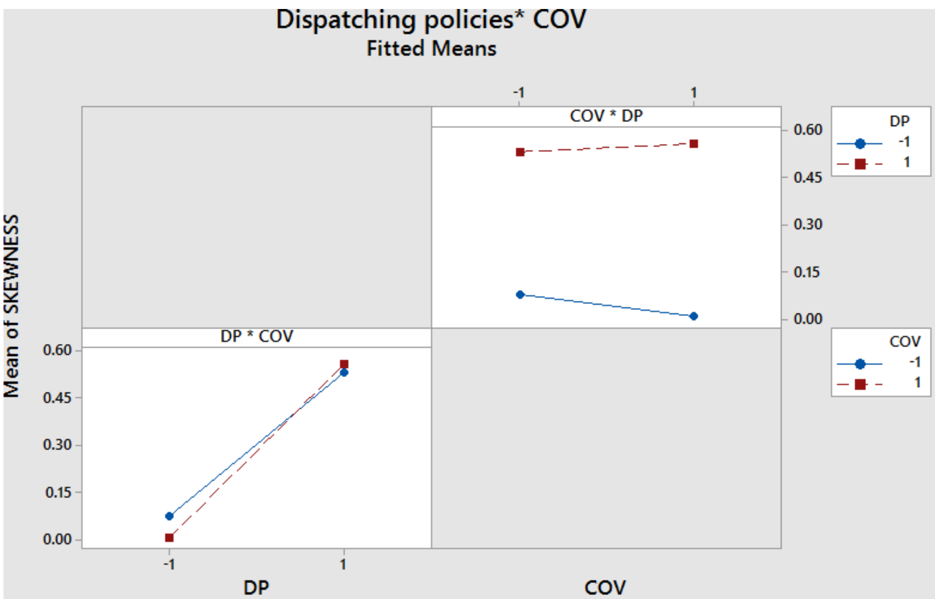


Figure 56. Two-way interaction between dispatching policies and COV for the skewness of the cycle time distribution for the mini-fab model.

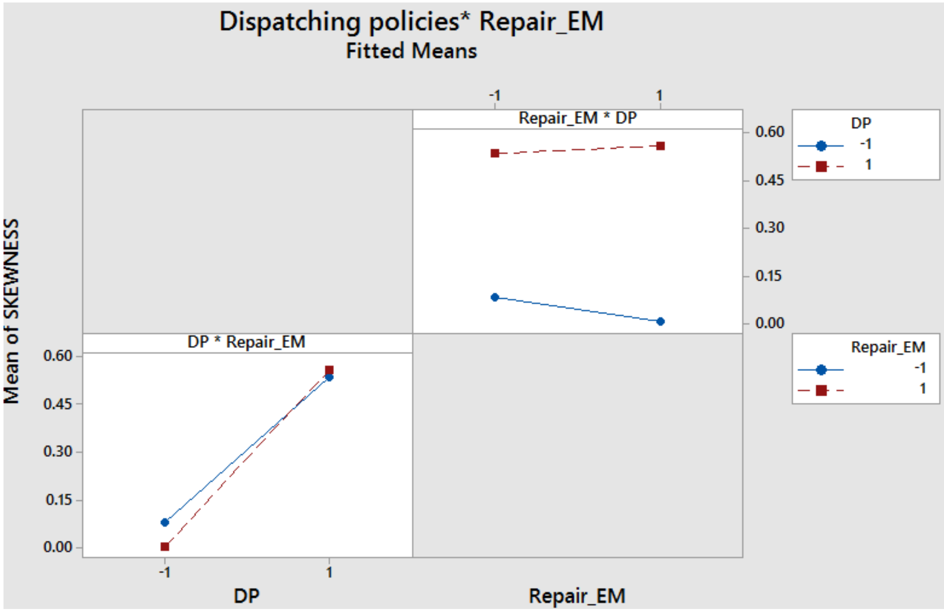


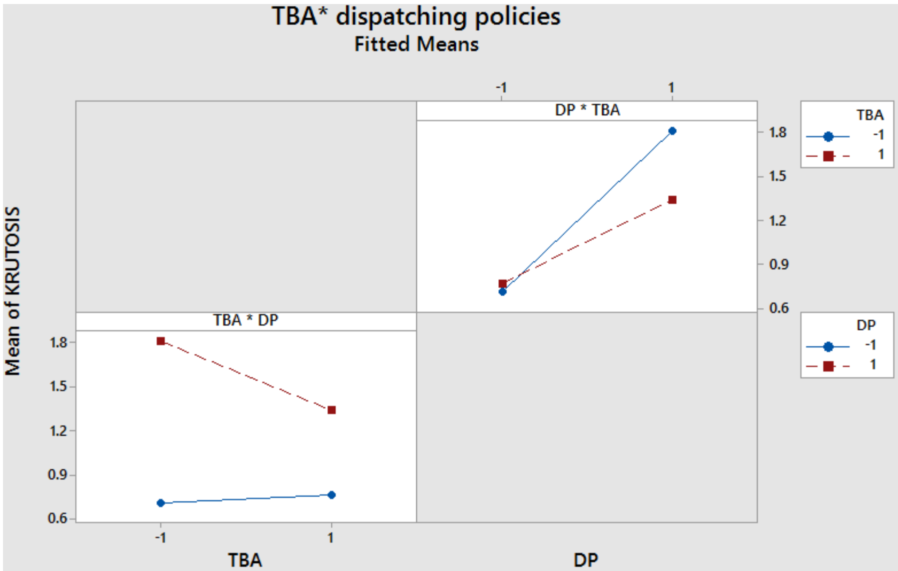
Figure 57. Two-way interaction between dispatching policies and mean time to repair in EM for the skewness of the cycle time distribution for the mini-fab model.

Considering the response variable of kurtosis, the main effect significant specifically to this is batch size. Table 28 lists the significant two-way interactions to kurtosis of cycle time distribution and the interactions are illustrated in Figures 58, 59 and 60 respectively.

Table 28. Two-way interactions significant specifically to kurtosis of cycle time distribution for the mini-fab model.

Two-way interactions	**Effect sizes**	**p-values**
TBA* dispatching policies	-0.13304	0.000
MPT* repair_EM	-0.1569	0.035
Dispatching policies* COV	0.01797	0.016

The interaction between TBA and dispatching policies as illustrated in Figure ***13***58 indicate that when the dispatching policies change from FIFO to SPT, the effect on kurtosis of cycle time distribution is more when TBA is at a lower value. The effect of change in the value of mean time to repair in EM on kurtosis of cycle time distribution is greater when MPT is at a higher value. This is demonstrated in Figure 59. Lastly, for the interaction between dispatching policies and COV as shown in Figure 60 shows that the effect of change of COV on kurtosis of cycle time distribution is greater when the



dispatching policies are SPT.

Figure 138. Two-way interaction between TBA and dispatching policies for kurtosis of the cycle time distribution for the mini-fab model.

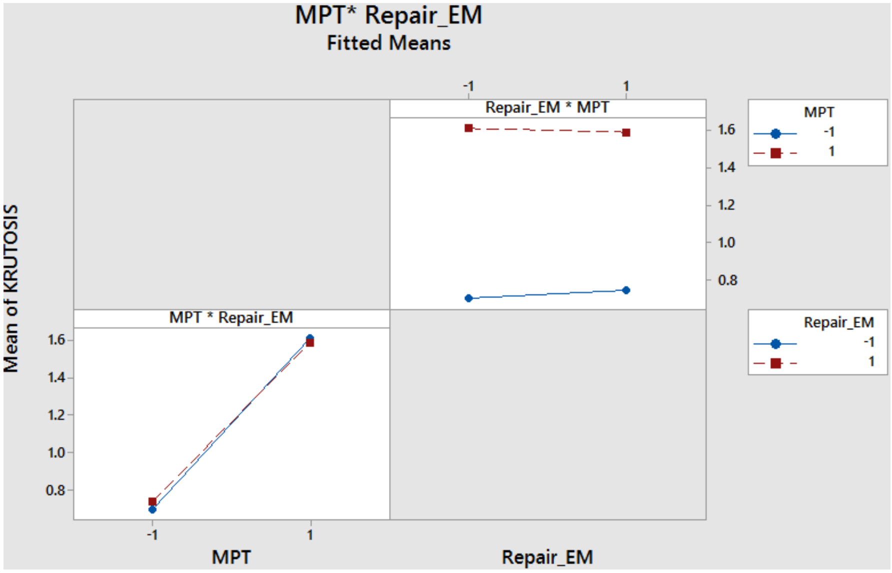


Figure 59. Two-way interaction between MPT and mean time to repair in EM for kurtosis of the cycle time distribution for the mini-fab model.

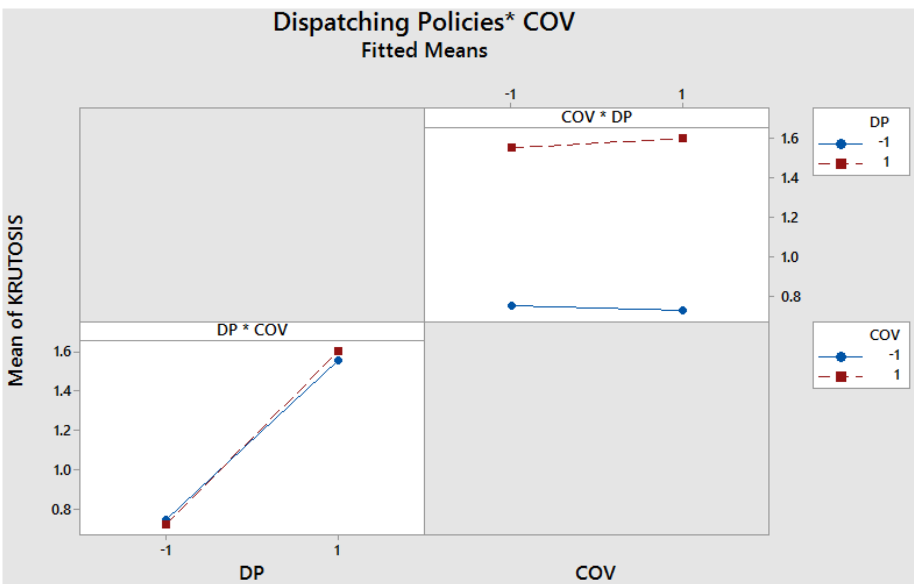


Figure 60. Two-way interaction between dispatching policies and COV for kurtosis of the cycle time distribution for the mini-fab model.

Moving ahead with the 25th percentile, the significant main effects are discussed in section 1.1. Table 29 lists the significant two-way interactions specific to the 25th percentile.

Table 29. Two-way interactions significant specifically to 25th percentile of cycle time distribution for the mini-fab model.

Two-way interactions	**Effect sizes**	**p-values**
TBA* batch size	0.1150	0.000
TBA* dispatching policies	0.1229	0.000
TBA* repair_EM	-0.0330	0.036
MPT* repair_EM	0.0333	0.035
Dispatching policies* repair_EM	-0.0330	0.036

The interactions between TBA and batch size, TBA and dispatching policies; and TBA and mean time to repair in EM are illustrated in Figure 61, Figure 62 and Figure 63 respectively. As seen in these plots, the effect of change of batch size, dispatching policies and mean time to repair in EM on the 25th percentile of the cycle time distribution is more noticeable when TBA is at a lower value. Second, the interaction between MPT and mean time to repair in EM illustrates that the change in mean time to repair in EM affects the 25th percentile of the cycle time distribution is greater at higher value of MPT. This is illustrated in Figure 64. Also, the change in mean time to repair in EM affects the 25th percentile of the cycle time distribution is more pronounced at dispatching policies FIFO as seen in Figure .

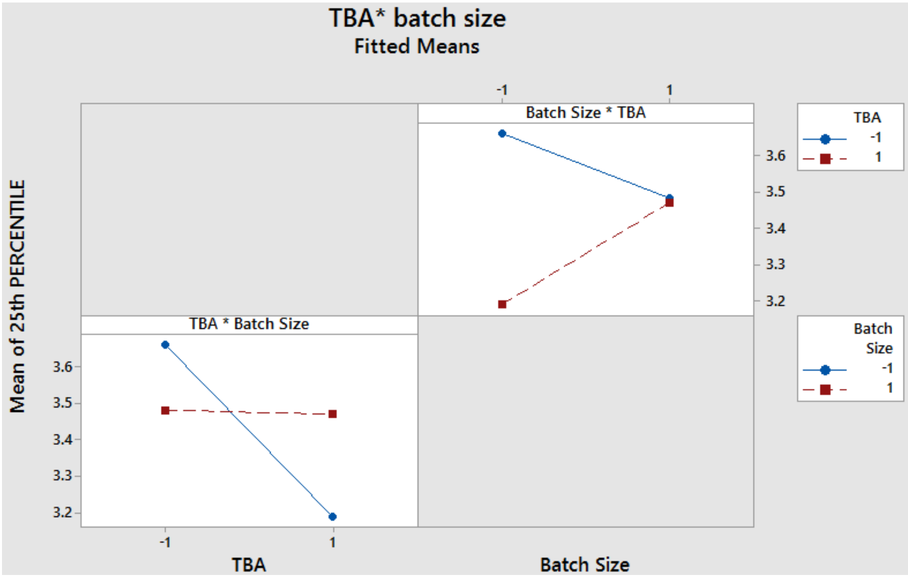


Figure 61. Two-way interaction between TBA and batch size for 25th percentile of the cycle time distribution for the mini-fab model.

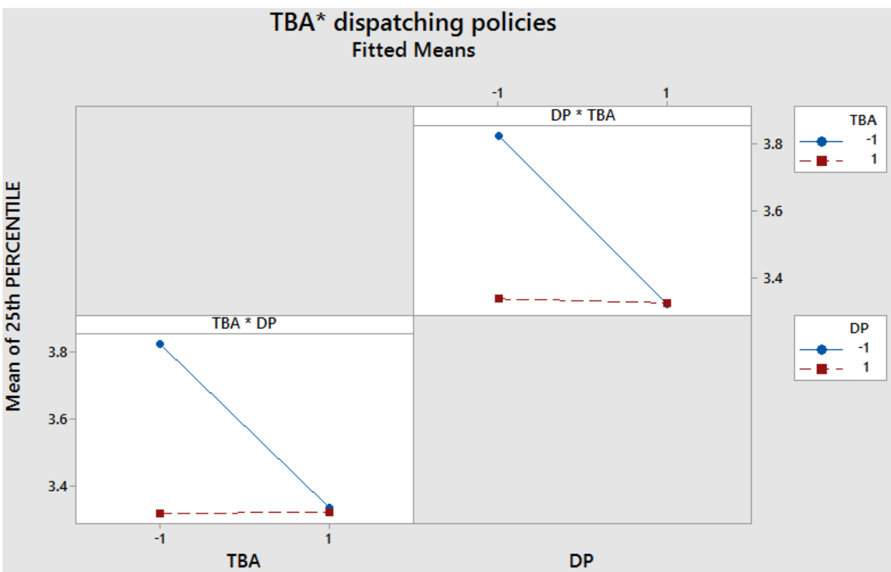


Figure 62. Two-way interaction between TBA and dispatching policies for 25th percentile of the cycle time distribution for the mini-fab model.

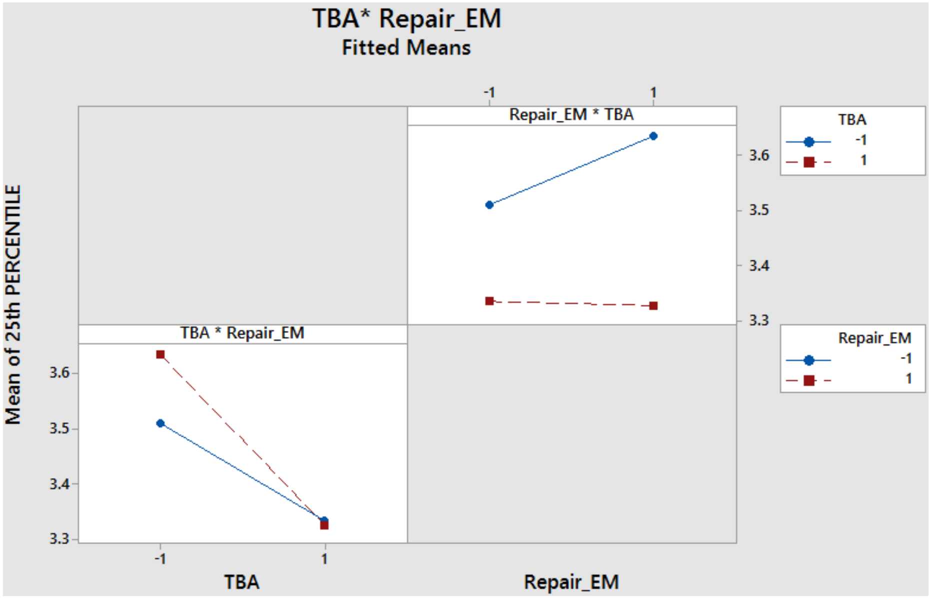


Figure 63. Two-way interaction between TBA and mean time to repair in EM for 25th percentile of the cycle time distribution for the mini-fab model.

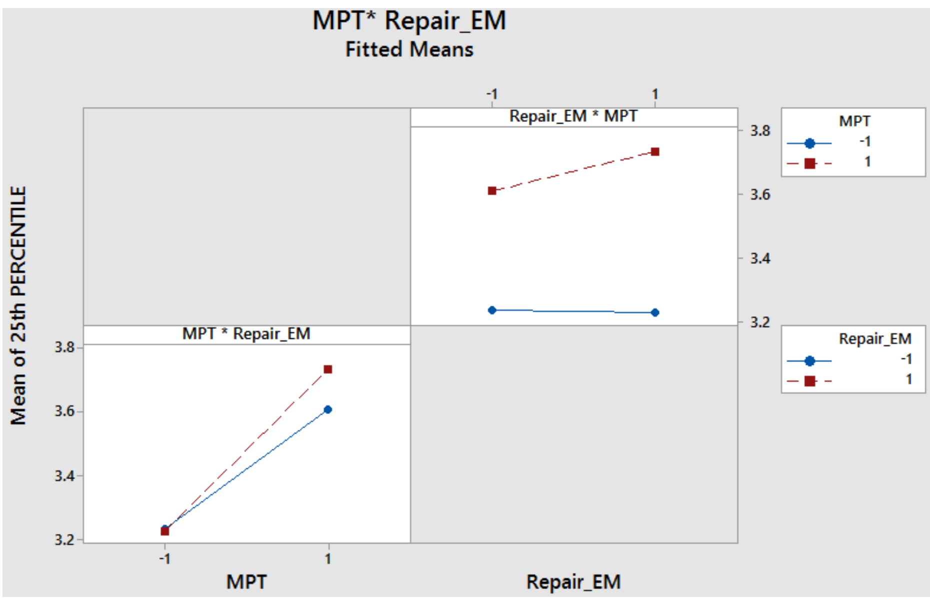


Figure 64. Two-way interaction between MPT and mean time to repair in EM for 25th percentile of the cycle time distribution for the mini-fab model.

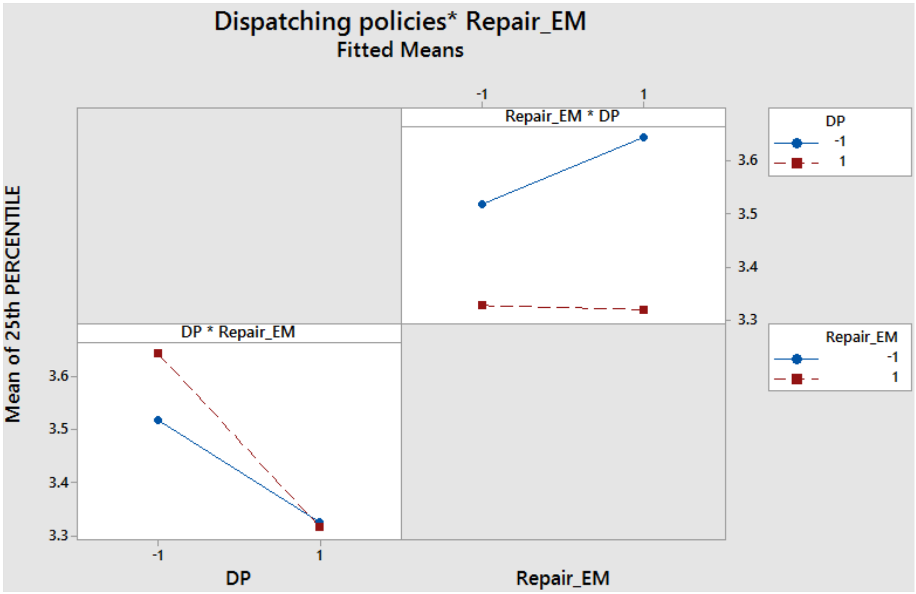


Figure 65. Two-way interaction between dispatching policies and mean time to repair in EM for 25th percentile of the cycle time distribution for the mini-fab model.

For the 50th percentile response variable, the interactions between TBA and batch size; and TBA and dispatching policies are statistically significant. Table 30 shows the effect sizes and p-values for these significant two-way interactions.

Table 30. Two-way interactions significant specifically to 50th percentile of cycle time distribution for the mini-fab model.

Two-way interactions	Effect sizes	p-values
TBA* batch size	0.1288	0.000
TBA* dispatching policies	0.1406	0.000

The two-way interaction between TBA and batch size; and TBA and dispatching policies are illustrated in Figure 66 and Figure 67 respectively. The interactions demonstrate that the effect of change in the levels of batch size and dispatching policies on 50th percentile of cycle time distribution is more pronounced when TBA is at a higher value.

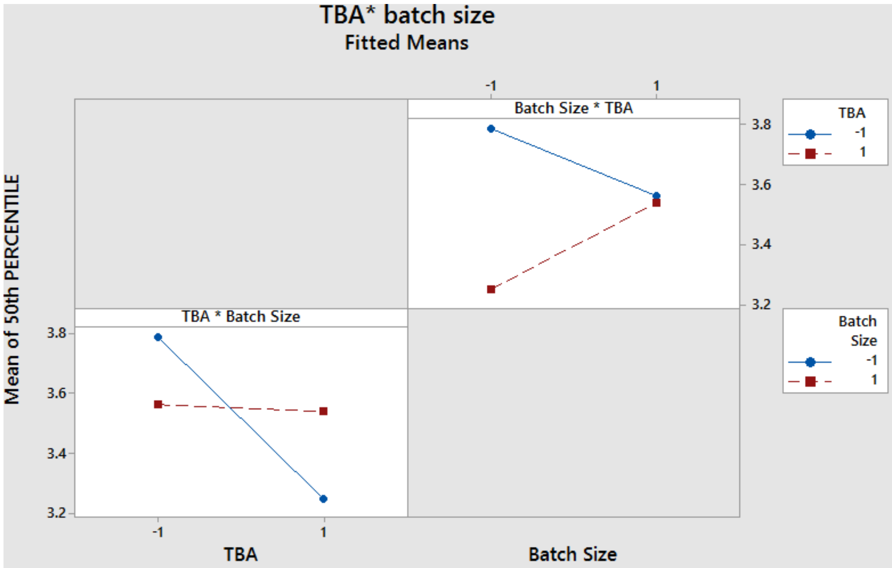


Figure 66. Two-way interaction between TBA and batch size for 50th percentile of the cycle time distribution for the mini-fab model.

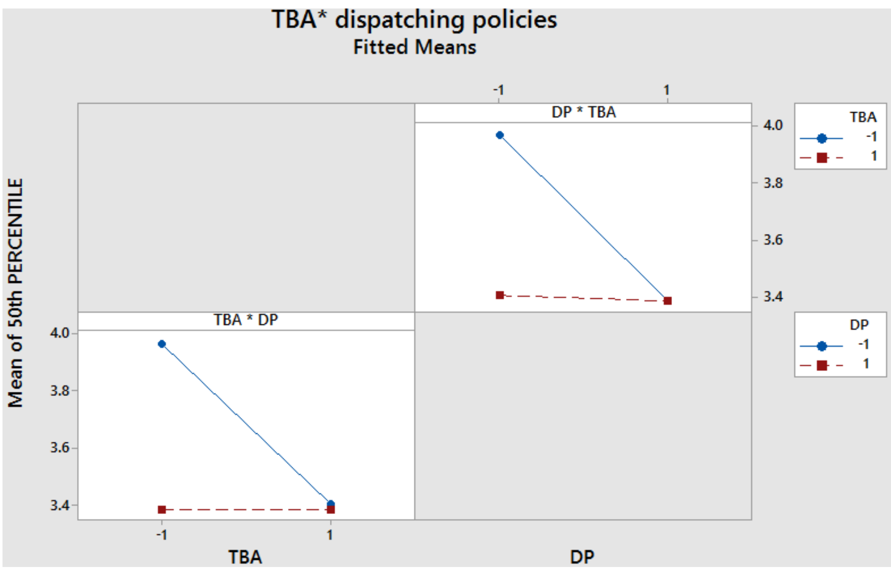


Figure 67. Two-way interaction between TBA and dispatching policies for 50^{th} percentile of the cycle time distribution for the mini-fab model.

Similar to the 50^{th} percentile of cycle time distribution, for the 75^{th} percentile of cycle time distribution, the interactions between TBA and batch size; and TBA and dispatching policies are significant. The effect size and p-values are listed in Table 31. The effect of the interaction plots for 75th percentile is similar as the 50th percentile of the cycle time distribution.

Table 31. Two-way interactions significant specifically to 75th percentile of cycle time distribution for the mini-fab model.

Two-way interactions	Effect sizes	p-values
TBA* batch size	0.1360	0.000
TBA* dispatching policies	0.1461	0.000

Finally, for the 95th percentile of cycle time distribution, the main effect of batch size is significant apart from the main effects mentioned in section 1.1. The significant two-way interactions are similar to that of 25th percentile. The effect sizes and p-values for these interactions are listed in Table 32.

Table 32. Two-way interactions significant specifically to 95th percentile of cycle time distribution for the mini-fab model.

Two-way interactions	Effect sizes	p-values
TBA* batch size	0.1444	0.000
TBA* dispatching policies	0.1059	0.000
TBA* repair_EM	-0.0257	0.041
MPT* repair_EM	0.0260	0.039
Dispatching policies* repair_EM	-0.0252	0.045

In order to determine if the results obtained from the analysis of the mini-fab model are also valid for the job-shop model, similar analysis of the results obtained from the DOE conducted using the job-shop was performed. The results are discussed next.

2. Results for job shop model

From the DOE conducted on the job shop model, the significant main effects, two-way and three-way interactions that were common across all the response variables at a significance of α ≤ 0.05 are listed in Table 33. The effect sizes and p-values for these significant main effects and interaction are listed in Table 33 and Table 34 respectively.

2.1 *Main Effects for job shop model common across all response variables:*

All the factors considered for the job shop model were found to be significant. Figure 68 summarizes the main effects for the job shop model.

	Mean	Standard Deviation	Skewness	Kurtosis	25th Percentile	50th Percentile	75th Percentile	95th Percentile
TBA	-0.607	-0.750	0.088	-0.063	-0.377	-0.392	-0.408	-0.463
MPT	0.888	1.256	0.068	0.309	0.553	0.617	0.668	0.826
DP	-0.148	0.126	0.552	0.733	-0.387	-0.415	-0.411	-0.259

Figure 68. Main effects of TBA, MPT and dispatching policies on all response variables for the job shop model.

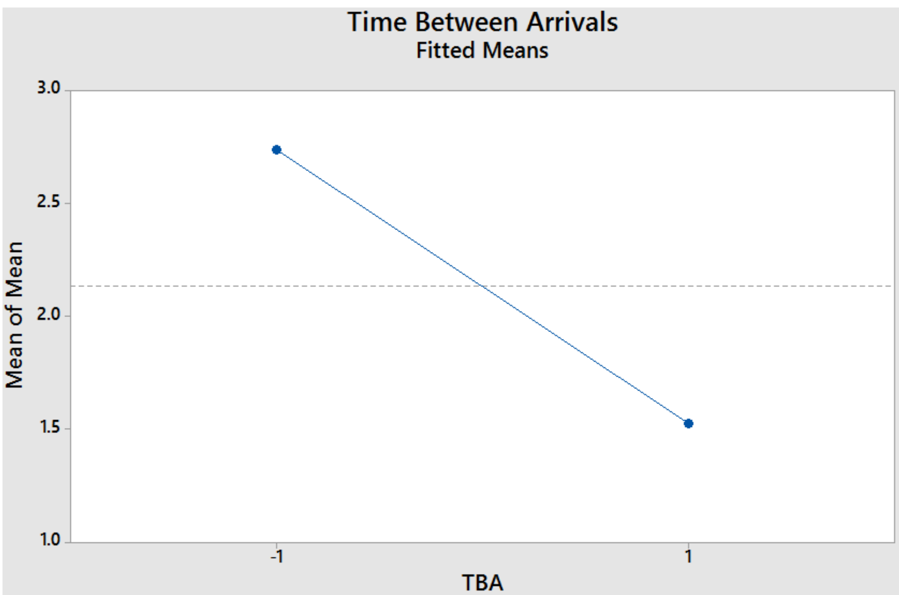
Table 33. Effect sizes for significant main effects and interactions for all the response variables for the job shop model.

	Effect Size							
	Mean	**Standard Deviation**	**Skewness**	**Kurtosis**	**25th Percentile**	**50th Percentile**	**75th Percentile**	**95th Percentile**
Main Effects and interactions								
TBA	-0.607	-0.750	0.088	-0.063	-0.377	-0.392	-0.408	-0.463
MPT	0.888	1.256	0.068	0.309	0.553	0.617	0.668	0.826
Dispatching Policies	-0.148	0.0321	0.552	0.733	-0.387	-0.415	-0.411	-0.259
MPT* Dispatching Policies	-0.144	0.099	0.352	0.411	-0.381	-0.409	-0.399	-0.255
TBA* MPT* Dispatching Policies	0.137	0.000	-0.242	-0.184	0.356	0.366	0.352	0.268

Table 34. p- values for significant main effects and interactions for all the response variables for the job shop model.

	p- values							
	Mean	**Standard Deviation**	**Skewness**	**Kurtosis**	**25th Percentile**	**50th Percentile**	**75th Percentile**	**95th Percentile**
Main Effects and interactions								
TBA	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
MPT	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Dispatching Policies	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
MPT* Dispatching Policies	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
TBA* MPT* Dispatching Policies	0.000	0.953	0.000	0.000	0.000	0.000	0.000	0.000

The effect of changing TBA is found to reduce the mean, standard deviation, kurtosis and the percentiles of the cycle time distribution. However, the skewness of the cycle time distribution increases as the TBA changes from a lower value to a higher value. The factor TBA affects the standard deviation of the cycle time distribution the most (effect size= 0.6065) and kurtosis of the cycle time distribution the least (effect size= 00632). Figure 69 and Figure 70 illustrate that the main effects of TBA on mean is more pronounced than skewness of the cycle time distribution indicted by the slope of the



lines connecting the two levels of TBA.

Figure 69. Main effects plot for TBA on mean of cycle time distribution for the job shop model.

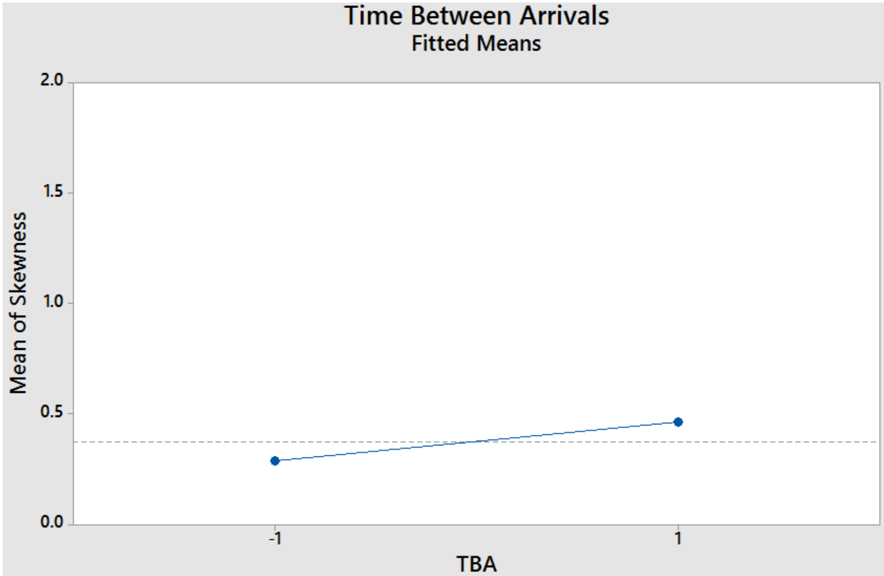


Figure 70. Main effects plot for TBA on skewness of cycle time distribution for the job shop model.

Figure 71 and Figure 72 illustrate the main effects of the factor MPT on both the standard deviation and 50th percentile of the cycle time distribution. These figures show that changes in MPT affects standard deviation the most and 50th percentiles the least. For both these responses, when MPT changes from a lower value to a higher value, the estimates of the cycle time distribution increase.

The change in dispatching policies affect the kurtosis of the cycle time distribution the most with effect size 0.733 and affect the standard deviation of the cycle time distribution the least with effect size of 0.126. Changing the dispatching policies from FIFO to SPT increases the standard deviation, skewness and kurtosis of the cycle time distribution but decrease the mean and the percentiles. As described earlier, this is because when the SPT dispatching rule is employed, the queue is ordered to prioritize jobs with shorter processing times of the tool. So, whenever a machine is freed, the entity in the queue with shortest processing time begins processing next. This in turn, means that entities with longer processing times are disadvantaged compared to FIFO dispatching policies, and may take a very long time to exit the system (Pinedo,1995). Figure **7147**3 and Figure 74 illustrate the main effects of dispatching policies on mean and skewness of the cycle time distribution respectively.

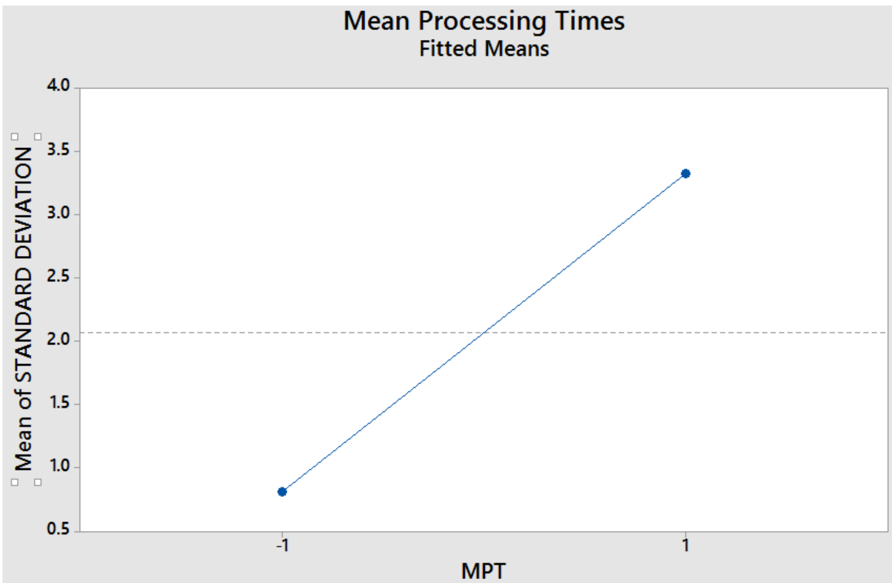


Figure 71. Main effects plot for MPT on standard deviation of cycle time distribution for the job shop model.

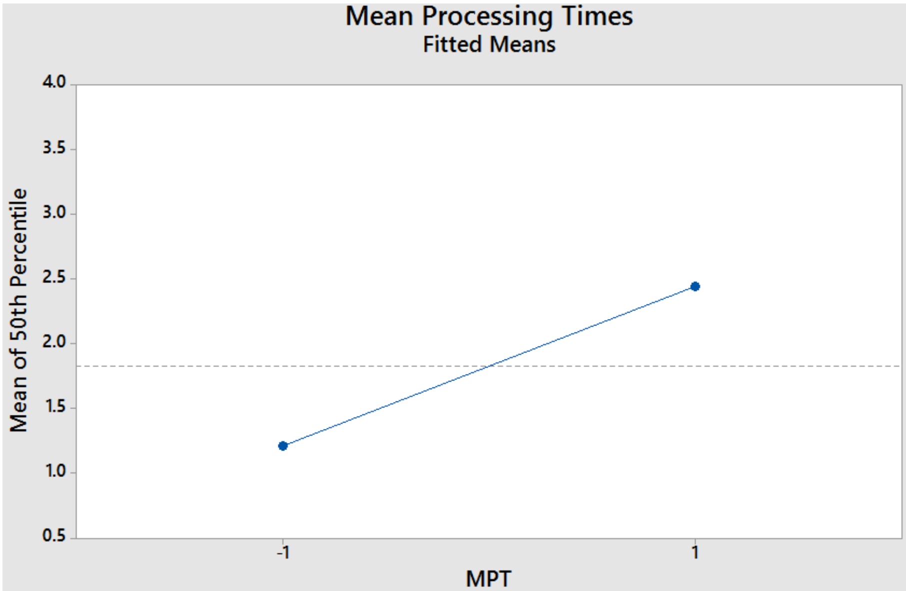


Figure 72. Main effects plot of MPT on 50th percentile of cycle time distribution for the job shop model.

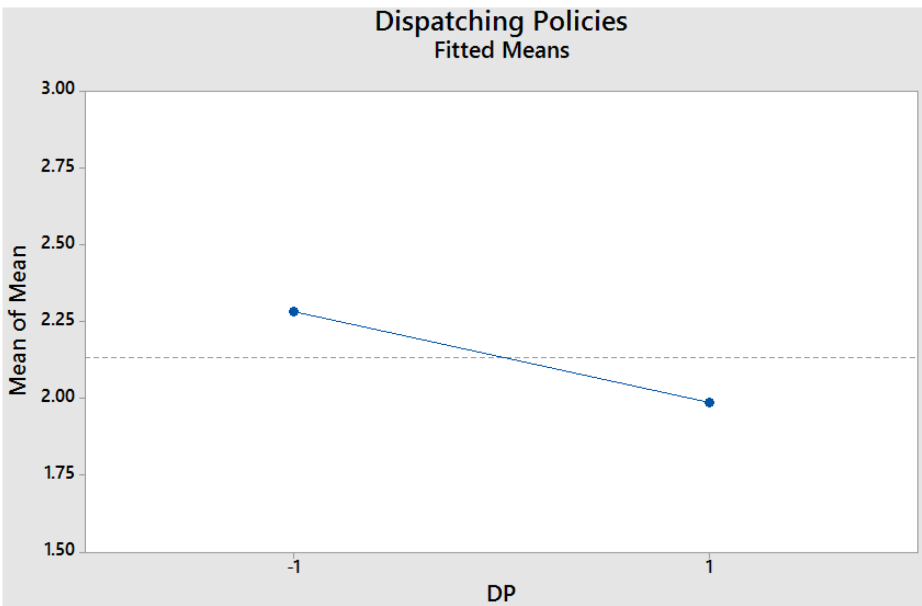


Figure 714. Main effects plot of dispatching policies on mean of cycle time distribution for the job shop model.

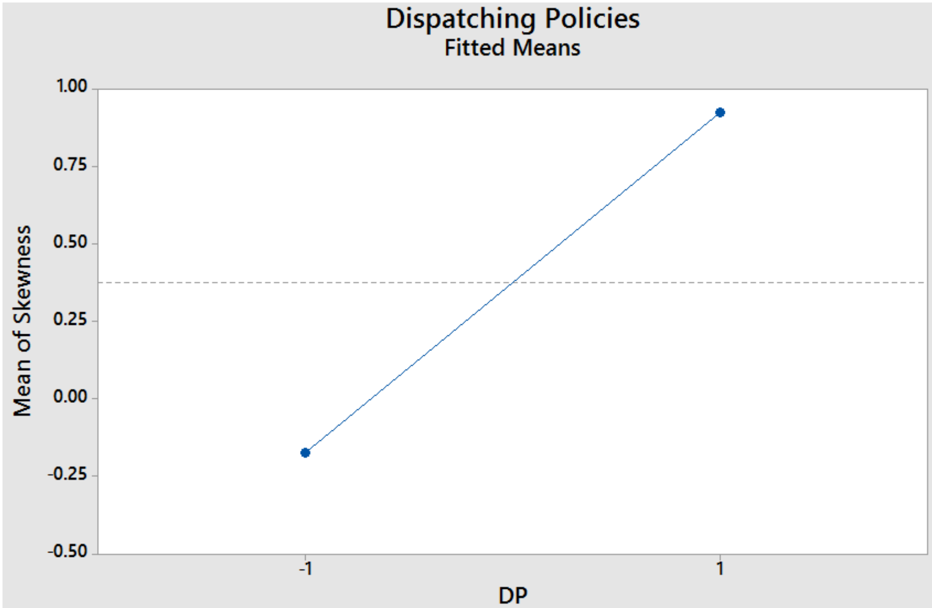


Figure 74. Main effects plot of dispatching policies on skewness of cycle time distribution for the job shop model.

2.2 *Two-way interactions for job shop model common across all response variables:*

The only significant two-way interaction for the job shop model is between MPT and dispatching policies. Figures 75 and 76 illustrate this interaction for mean and skewness of cycle time distribution respectively.

The interaction between MPT and the dispatching policies demonstrates that the influence of changing the dispatching policies from FIFO to SPT is more pronounced when MPT is at a higher value. This is true for all the response variables. When dispatching policies are changed from FIFO to SPT, the system gets congested. In such a scenario, the effect of MPT at a higher value will increase the pressure on system and is hence more significant than when MPT is at a lower value.

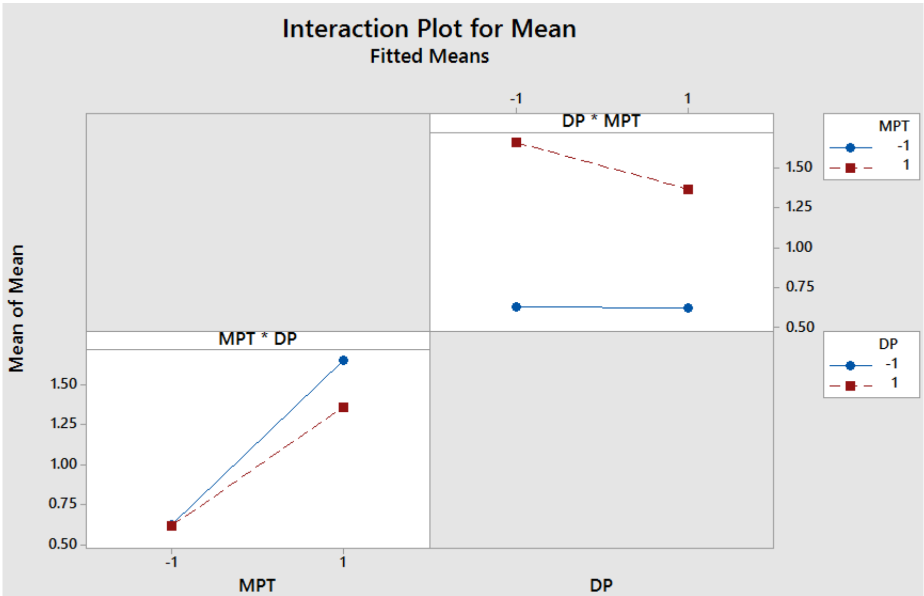


Figure 75. Interaction plot of MPT and dispatching policies on mean of cycle time distribution for the job shop model.

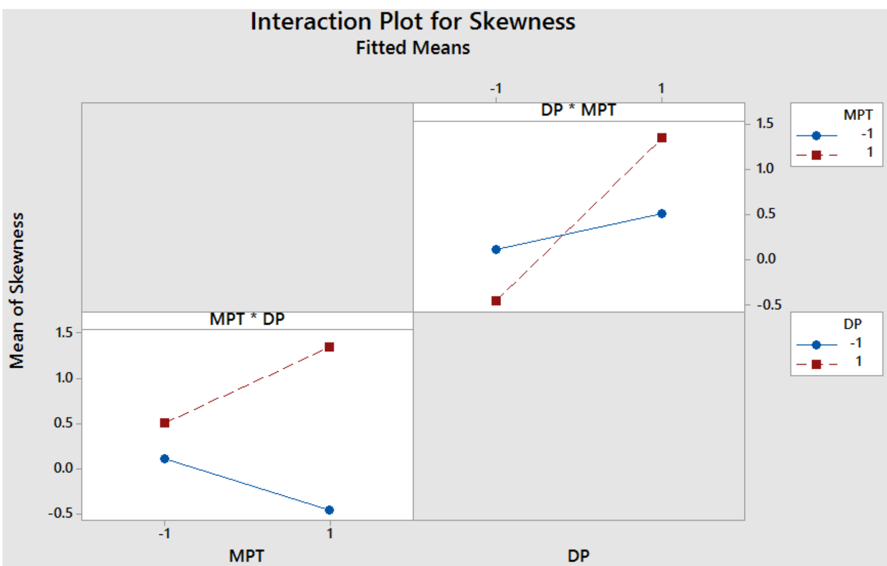


Figure 76. Interaction plot of MPT and dispatching policies on skewness of cycle time distribution for the job shop model.

2.3 ***Three-way interactions for Job shop model common across all response variables:***

The three-way interaction between TBA, MPT and dispatching policies are significant for mean, skewness, kurtosis and the percentiles of cycle time distribution. As discussed earlier in this chapter, first, the simple two-way interactions between MPT and dispatching policies were tested at the two levels of TBA. The p-values are listed in Table 35. The interaction is found to be significant for all the significant response variables of the cycle time distribution for both the levels of TBA. Second, the interaction between TBA and dispatching policies were tested at the two levels of MPT. The p-values for this interaction are listed in Table 36. The interaction is found to be significant for all the significant response variables of the cycle time distribution for both the levels of MPT. Finally, when the interaction between TBA and MPT was tested at the two levels of dispatching policies, it is found to be significant for all the significant response variables of the cycle time distribution. The p-values are listed in Table 37.

Table 35. Simple two- way interaction between MPT and dispatching policies at levels of TBA for all response variables for the job shop model.

Response Variables	Levels of TBA	p- value
Mean	EXPO (225) mins EXPO (260) mins	0.000 0.000
Skewness	EXPO (225) mins EXPO (260) mins	0.000 0.000
Kurtosis	EXPO (225) mins EXPO (260) mins	0.000 0.000
25th Percentile	EXPO (225) mins EXPO (260) mins	0.000 0.000
50th Percentile	EXPO (225) mins EXPO (260) mins	0.000 0.000
75th Percentile	EXPO (225) mins	0.000

	EXPO (260) mins	0.000
95th Percentile	EXPO (225) mins EXPO (260) mins	0.000 0.000

Table 36. Simple two- way interaction between TBA and dispatching policies at levels of MPT for all response variables for the job shop model.

Response Variables	**Levels of MPT**	**p- value**
Mean	-30% of mean +30% of mean	0.000 0.000
Skewness	-30% of mean +30% of mean	0.000 0.000
Kurtosis	-30% of mean +30% of mean	0.000 0.000
25th Percentile	-30% of mean +30% of mean	0.000 0.000
50th Percentile	-30% of mean +30% of mean	0.000 0.000
75th Percentile	-30% of mean +30% of mean	0.000 0.000
95th Percentile	-30% of mean +30% of mean	0.000 0.000

Table 37. Simple two- way interaction between TBA and MPT at levels of dispatching policies for all response variables for the job shop model.

Response Variables	**Levels of dispatching policies**	**p- value**
Mean	FIFO SPT	0.000 0.000
Skewness	FIFO SPT	0.000 0.000
Kurtosis	FIFO SPT	0.000 0.000
25th Percentile	FIFO SPT	0.000 0.000
50th Percentile	FIFO SPT	0.000 0.000
75th Percentile	FIFO SPT	0.000 0.000
95th Percentile	FIFO	0.000

	SPT	0.000

Some other two-way interactions were also found to be significant for specific response variables. These interactions are discussed next.

2.4 *Two-way interactions significant to a subset of the responses variable for the job shop model:*

First, the interaction between TBA and MPT; and between TBA and dispatching policies are significant for mean, skewness and the percentiles of the cycle time distribution. The effect sizes and p-values are listed in Table 38 and Table 39 respectively.

Table 38. Effect sizes of two-way interaction between TBA and MPT and between TBA and dispatching policies for the job shop model.

Two-way interactions	**Mean**	**Skewness**	**25th percentile**	**50th percentile**	**75th percentile**	**95th percentile**
TBA* MPT	-0.590	0.133	-0.366	-0.383	-0.393	-0.437
TBA* dispatching policies	0.138	-0.255	0.354	0.368	0.354	0.268

Table 39. p- values of two-way interaction between TBA and MPT and between TBA and dispatching policies for the job shop model.

Two-way interactions	**Mean**	**Skewness**	**25th percentile**	**50th percentile**	**75th percentile**	**95th percentile**
TBA* MPT	0.000	0.000	0.000	0.000	0.000	0.000
TBA* dispatching policies	0.000	0.000	0.000	0.000	0.000	0.000

The interaction plots of two-way interactions between TBA and MPT and between TBA and dispatching policies for the mean of the cycle time distribution are illustrated in Figure 77 and Figure 78. These interactions illustrate that the effect of changing TBA from a lower value to a higher value is more pronounced when MPT is at a higher value and when dispatching policies are set to FIFO. This is also true for the percentiles of the cycle time distribution.

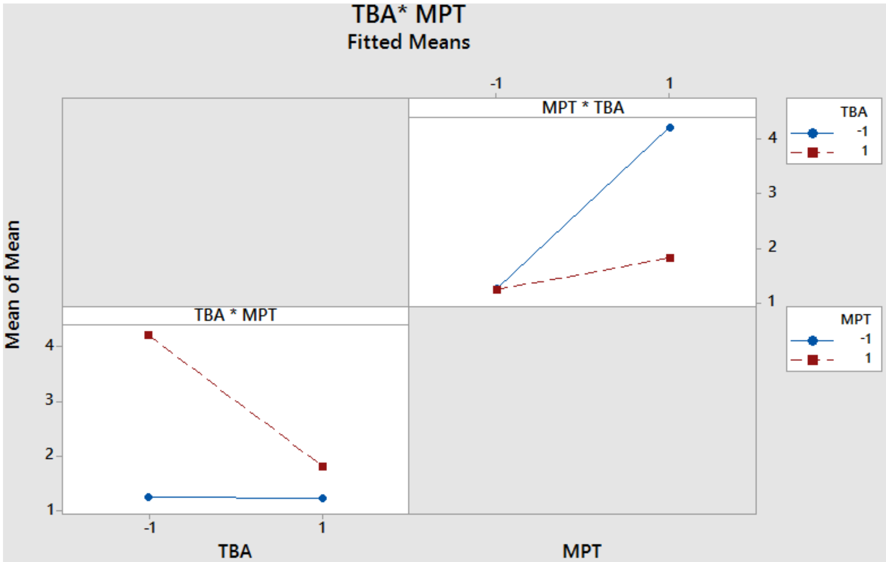


Figure 77. Interaction plot between TBA and MPT for the mean of cycle time distribution for the job shop model.

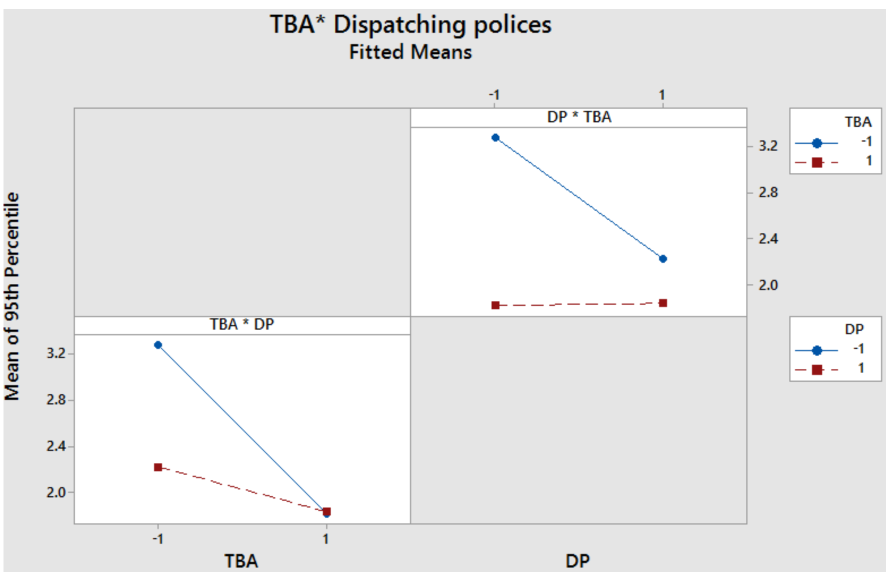


Figure 78. Interaction plot between TBA and dispatching policies for the mean of cycle time distribution for the job shop model.

Figure 79 and Figure 80 illustrate the two-way interaction between TBA and MPT; and between TBA and dispatching policies for standard deviation of cycle time distribution. These interactions show that the effect of increasing TBA from a lower value to a higher value is more prominent when MPT is at a higher value and dispatching policies are at SPT. This is also true for skewness and kurtosis of the cycle time distribution.

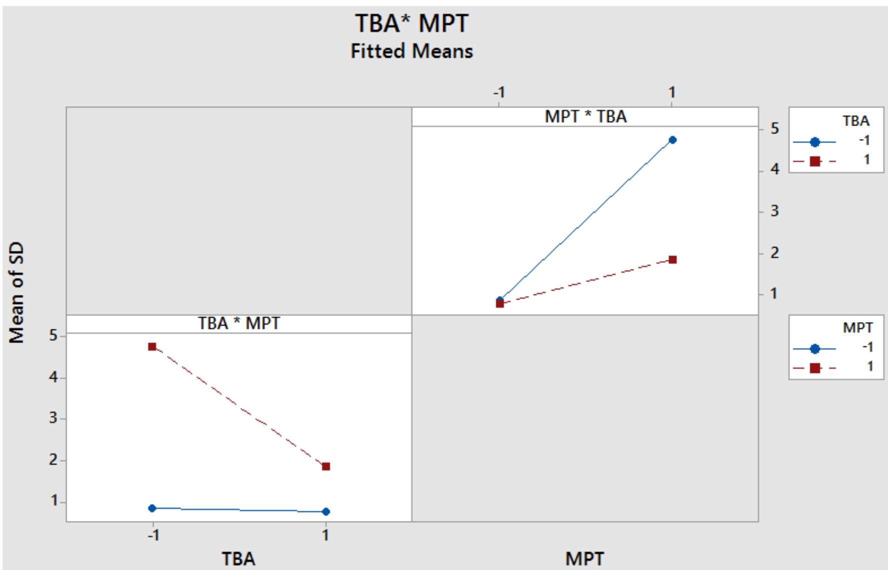


Figure 79. Interaction plot between TBA and MPT for standard deviation of cycle time distribution for the job shop model.

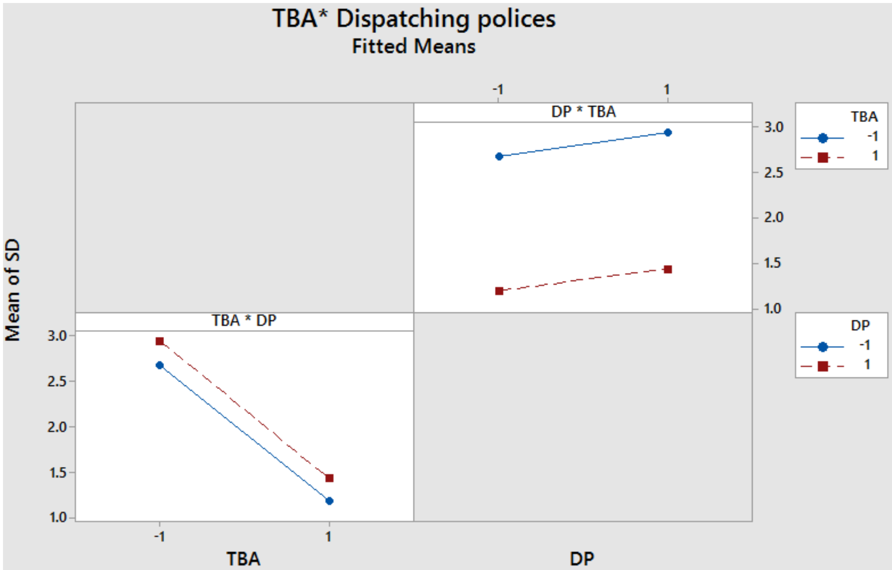


Figure 80. Interaction plot between TBA and dispatching policies for standard deviation of the cycle time distribution for the job shop model.

The next chapter discusses the implications of the results discussed in this chapter. Chapter 4 also discusses the limitations and future work to this research work.

4. Discussions and Conclusions

This chapter is divided into four sections. In the first section, the key points from the results are discussed. In the second section, the similarities between the mini-fab model and the job shop model are discussed. The third section gives three example scenarios, each of which describes potential changes to a production line. Incorporating the findings from Chapter 3, the scenarios illustrate how the findings from this work could be utilized by a decision maker to understand how potential changes to the production line would influence the cycle time distribution and associated delivery date quotations. Finally, the limitations of the work and directions for future work are described.

1. Summary of key findings from mini-fab and job shop models

Some of the key findings and intuitions from the experimentation conducted using both the mini-fab and job shop simulation models are provided next:

- If all other things in the production facility remain approximately the same, when the start rate (i.e., the rate at which new jobs are started in the production facility) increases, the mean time as well as the 25^{th}, 50^{th}, 75^{th}, and 95^{th} percentile of the distribution increases. Start rate is the inverse of TBA. Jobs are generally taking longer to get through the production facility. Additionally, both the variability and the kurtosis of the distribution surrounding the cycle time have shown to increase. Together, these results imply that, compared to systems in which the start rate is lower, a greater percentage of jobs take a comparatively small or large amount of time to complete. The skewness of the cycle time distribution reduces when the start rate increases for the job shop model. The reason for reduction of skewness can be assumed to be because of the absence of batching in the job shop model.

- In two-way interactions involving start rate, the effect of changing the other factor is more pronounced at higher start rates. This is explained by the fact that when start rate is more, the system is more congested, making the queue longer thus increasing the utilization of the bottleneck tool group (Akcali et al., 2001; Meidan et al., 2011).
- When the mean processing time of the tool groups increases, as expected, the mean and the 25^{th}, 50^{th}, 75^{th} and 95^{th} percentiles of the cycle time distribution increases. Also, the variability, skewness and kurtosis of the cycle time distribution have shown to increase.
- In two-way interactions involving mean processing times of all tool groups, the effect of changing another factor is more pronounced when mean processing time is higher. This is observed in both mini-fab and job shop model. When the machines take longer time to process, the queue is longer. Mean processing time is an important factor in determining the cycle time (Meidan et al., 2011).
- Changing the dispatching policies from FIFO to SPT have shown to decrease the mean and the 25^{th}, 50^{th}, 75^{th} and 95^{th} percentile. However, the variation, skewness and kurtosis of the distribution surrounding the cycle time increases.
- When one of the factors in a significant two-way interaction is batch size, the effect of the change of the other factor is more noticeable when batch size is smaller. When processing of parts is done in smaller batches, more parts wait to be batched thus increasing the queue of the system (Qi et al., 2002).

2. Similarities between the two systems

The mini-fab model is a semiconductor system that includes complexities such as batching, loading, and sequence dependent set ups. The job shop model on the other hand is a much simpler system that does not include these same complications. The impact of these differences can be seen on the resulting cycle time distributions, as shown in Figures 6 and 9. A goal of this research was to understand whether the findings about the relationship between controllable factors in the production system and parameters describing the cycle time distribution are generalizable across the two systems. If they are, it lends support to the findings as being transferable outside the constraints of just a single model. The findings in Chapter 3 demonstrate that, despite substantial differences in the complexities between these system, there are similarities in the way the controllable factors influence the cycle time distribution. These similarities are detailed next.

In both systems, changing the start rate affects the standard deviation of the cycle time distribution more than any of the parameters describing the cycle time distribution. As the start rate decreases, the parts begin manufacture in the system at a slower rate. Since the inter arrival rate is larger, the rate at which the parts exit the system will also be larger. Correspondingly, the variation in the distribution of the cycle time also increases. Tables 10 and 33 demonstrate that the impact of change in start rate on the outcome variables in mini-fab model are relative to that of the job shop model. The effect of increased start rate is more pronounced for the upper tails of the distribution. For example, in both the systems, from Chapter 3, the effect size of change in start rate on 50^{th} percentile is more than that on the 25^{th} percentile and so on.

The effect of change in mean time for processing is the same on all the outcome measures for the mini-fab and the job shop model. In both the systems, the influence of change of mean processing time is the greatest on the standard deviation of cycle time distribution. As with the previous example related to start rate, the effect of increasing mean processing time is more pronounced for the upper tails of the cycle time distribution. furthermore, the magnitude of the effect sizes of increasing mean processing time on the percentiles of the cycle time distribution for the mini-fab model are relative to that of the job shop model. Also, the effect of changing dispatching policies from FIFO to SPT on the response variables for the job shop model are similar to that for the mini-fab model.

The two-way interaction between mean processing time and dispatching policies is significant to both the mini-fab model and the job shop model. The impacts of the change in the dispatching policies from FIFO to SPT is more pronounced when the time taken by the tool groups to process is more. As illustrated in Figures 17 and 68, the effect of this interaction on the response variables in the mini-fab model and the job shop model are similar.

Based on these similarities in findings between the two systems, we argue that for systems bounded by the distributions of the cycle time of the mini-fab model and the job shop model, the described effects can be predicted to be similar. These similarities make the findings from mini-fab transferable to other systems and, correspondingly, can save a considerable amount of time and resources that, without the findings presented here, would be necessary for delivery date estimation.

3. Implications of the findings to practitioners

One of the goals of any semiconductor or high volume electronics manufacturing facility is to predict cycle time (and so customer delivery times) more accurately. The work in this thesis is intended to aid decision makers with this task. To illustrate this in a context relevant to practitioners, three scenarios are presented. In each scenario, a hypothetical change to a controllable aspect of a semiconductor manufacturing system is given. Incorporating the findings from Chapter 3, the expected implications to the production system and to the quotation of delivery dates of the potential change to the system are discussed and explained.

Case 1. A new parallel machine for the bottleneck tool group is purchased.

In a production facility, the bottleneck tool group is the most heavily utilized tool group and, therefore, the tool group that controls and limits the throughput. Also, the bottleneck tool group is typically the most expensive piece of equipment in the facility (Akcali et al., 2001). Assume that a production facility is considering the purchase of a new tool to replace the bottleneck station in their facility. It is expected that the new tool would have a mean processing time 10% faster than the current tool. While the decision makers know that an increase in the production rate at the bottleneck tool is expected to have a positive influence on the mean time in system for jobs in the facility, in an effort to determine whether the cost of the new tool is warranted, he is also interested in understanding the other expected implications to the cycle time distribution.

Common across all response variables, mean processing time interacts with start rate, batch size and dispatching policies. Drawing from the findings from Chapter 3, the effect of reduction in mean processing time is more noticeable when the start rate is higher. The interaction between mean processing time and start rate has a larger impact on the standard deviation of the cycle time distribution and a smaller impact on the skewness of the cycle time distribution. This is illustrated in Figure 25. This means that when the mean time to process is reduced, by having a start rate higher, the due date will be lesser. This also means that for the same cycle time, the manufacturer can now produce more products. However, since the standard deviation increases, the variation in predicting the cycle time will be more.

From the findings in Chapter 3, a decrease in mean processing time is more pronounced when the batch size is smaller. This interaction influences the standard deviation of the cycle time distribution the most and the kurtosis of the cycle time distribution the least. This implies that the practitioner will have more confidence in quoting the delivery dates. Figure 30 shows that the effect of reducing mean processing time in the interaction between mean processing time and dispatching policies is more evident when dispatching policies are FIFO for the mean and the percentiles of cycle time distribution. However, the impact of reducing mean processing time is more evident when dispatching policies are SPT for standard deviation, skewness and kurtosis of cycle time distribution. This implies that when dispatching policies are set at FIFO, delivery dates estimates will be smaller when the percentiles are smaller due to installation of the new machine. Also, the variability in delivery dates will be lesser when the dispatching policies are set at SPT by which the practitioner will now have more confidence in quoting delivery dates.

The two-way interaction between mean processing time and coefficient of variance of unloading is significant for mean and standard deviation of the cycle time distribution. Figures 42 and 50 demonstrate that the effect of lowering mean processing time is more pronounced when coefficient of variance of unloading is a higher percentage of mean. From these figures, it can be said that reduction in mean processing time when the coefficient of variance at unloading is higher, the time taken to complete the products will be lesser as the means and standard deviation reduces. The two-way interaction between mean processing time and mean time to repair in emergency failures is significant for mean, standard deviation, skewness, kurtosis, 25th percentile and 95th percentile of the cycle time distribution. The effect of installing a new machine will be more prominent on the mean, standard deviation and the percentiles when the mean time to repair in emergency failures is high and more prominent on skewness and kurtosis of the cycle time distribution when the mean time to repair in emergency failures is low. This means that when the mean time to repair in EM is lesser, the delivery dates is higher.

Case 2. Dispatching policies are changed from FIFO to SPT.

By changing the dispatching policies from FIFO to SPT, the parts that have higher processing times take a very long time to exist the system. The two-way interactions that are significant across all the response variables when dispatching policies are changed are between dispatching policies and batch size; and dispatching policies and mean processing time.

For all the response variables the effect of changing dispatching policies from FIFO to SPT is more pronounced when batch size small and when mean processing time is high. This is indicated in Figures 24 to 26. From Table 10, this interaction is shown to influence the 75th percentile the most and standard deviation the least. The increase in the percentiles would make delivery dates to be longer since the distribution now has longer tails.

Figure 24 illustrates that the interaction between dispatching policies and mean processing time influences kurtosis the most and standard deviation the least. When mean processing time is at a higher value and when the dispatching policies are changed to SPT from FIFO, the 75th percentile reduces making the delivery date quotations to be shorter.

The two-way interaction between dispatching policies and start rate is significant for the mean, skewness, kurtosis and the percentiles of cycle time distribution. For this interaction, the effect of change in dispatching policies from FIFO to SPT will be evident when the start rate of the parts is more. By changing the dispatching policies, the effect of start rate reduces, and this reduces the estimates of the cycle time. Also, the two-way interaction between dispatching policies and coefficient of variance of unloading is significant for skewness and kurtosis of cycle time distribution and the effect is noticeable when coefficient of variance of unloading is high. So, when %COV is higher, for the same cycle time, more products can be manufactured. Finally, the two-way interaction between dispatching policies and mean time to repair in emergency failures is significant for mean, standard deviation, skewness, 25^{th} percentile and 95^{th} percentile of cycle time distribution. When the dispatching policies are changed from FIFO to SPT, at higher values of mean time to repair, the delivery dates would get shorter.

Case 3. Implementing robots for loading, unloading and machine set ups.

When robots are implemented, the variation in time for loading, unloading and machine setups is smaller. In such a case, the two-way interaction between start rate and coefficient of variance of unloading; and between mean processing time and coefficient of variance of unloading should be examined (i.e., those are the statistically significant interactions related to the coefficient of variance at setups). The impact of reducing the coefficient of variation of unloading will be more pronounced when the start rate of the parts is high and when the mean processing time is high. The two-way interaction between batch size and coefficient of variance of unloading significant for standard deviation of cycle time distribution demonstrates that by reducing the coefficient of variance of unloading will be more noticeable when the batch size is smaller. Also, the two-way interaction between dispatching policies and coefficient of variance of unloading is significant for skewness and kurtosis of the cycle time distribution and it indicates that when the coefficient of variance of unloading reduces, the effect is more prominent when dispatching policies are at SPT. When the coefficient of variance at unloading is reduced, the estimates of the cycle time distribution are reduced when start rate is higher, mean processing time is higher and when batch size is smaller. In such cases, the due dates will be smaller (vs when the coefficient of variance at unloading is higher). Hence, quoting delivery dates will also be more accurate.

4. Limitations and Future work

Future work in this area will include several important topics. In order to understand the future work, the limitations of this work is discussed first. One of the limitations for this work is that the job shop model does not include batch processing, loading, unloading etc. The factors common to the mini-fab and the job shop model include start rate, mean processing time and dispatching policies. A generalization of the effects of change in these factors is done on the basis of comparison. However, no such generalization for other factors significant to the mini-fab model which include batch size, coefficient of variance of unloading and mean time to repair in emergency failures can be done. A second limitation of this work includes that the sensitivity analyses done in the early stages of this work eliminated many factors that affect the cycle time distribution. As mentioned in Chapter 2, from the pilot runs, if the values of the outcome measures changed by 10% from the baseline values, the cycle time distribution was considered to be sensitive to the factor. However, there is no rule of thumb for selecting the value of the percent change for the outcome measures from the baseline values. As a result, it is possible that factors that really do impact the cycle time distribution were excluded from this analysis.

Future work will include in building a job shop model that consists of these factors. A similar DOE can be run in order to analyze the effects of these factors for the job shop model. Similar to the comparison explained in this chapter, generalization for these factors can also be done. Based on this work, a generic function can be developed to predict the cycle time of the systems so that estimating delivery dates is simpler. Also, different type of manufacturing systems with different cycle time distributions should be considered to have a better understand the generalization of the findings.

Another future work will include in considering all the factors including the ones found not sensitive in this work (for example, percentage of rework done) and conducting similar analyses. The impact of change of these factors can be understood. The simulation runs give accurate results but are time consuming. Another future work will include using other models to predict the cycle time and compare the results obtained in this work.

Finally, for the interactions found to be significant, in order to know the levels of the factors for which the interaction impacts the outcome measures, a pairwise comparison can be done. This would allow a decision maker to have a better understanding of the interactions. For example, while examining the interaction between start rate and mean processing time, from the findings from Chapter 3, it is clear that an increase in start rate is more pronounced when the mean processing time is higher. By conducting a pairwise comparison, a researcher could understand whether there is a statistical difference between the case where mean processing time is high with high start rates and the case where mean processing time is low with high start rates.

REFERENCES

Akcali, E., Nemoto K. and Uzsoy, R. (2001) Cycle-Time Improvements for Photolithograph Process in Semiconductor Manufacturing, *IEEE Transactions on Semiconductor Manufacturing*, **14**(1), 48-56.

Akenmann, B.E., Bekki, J.M., Fowler, J., Mackulak, G.T., Nelson, B.L. and Yang, F. (2011) Simulation in Production Planning: An Overview with Emphasis on Recent Developments in Cycle Time Estimation, *Planning production and Inventories in the Extended Enterprise: A State of the Art Handbook*, Volume1, Springer Science, New York.

Backus, P., Janakiram, M., Mowzoon, S., Runger, G.C. and Bhargava, A. (2006) Factory cycle-time prediction with a data-mining approach, *IEEE Transactions on Semiconductor Manufacturing*, **19**(2), 252-258.

Banks, J., Carson, J.S. and Nelson, B. (1996) *Discrete-Event System Simulation,* Second edition, Prentice Hall, Upper Saddle River, NJ.

Boyaci, T. and Ray, S. (2006) The Impact of Capacity Costs on Product Differentiation in Delivery Time, Delivery Reliability, and Price, *Production and Operations Management,* **15**, 179-197.

Chang, S. and Huang, H (2002) Cycle time estimation for wafer fab with engineering lots, *IIE Transactions*, **34**, 105-118.

Chang, P.C. and Liao, T.W. (2006) Combining SOM and fuzzy rule base for flow time prediction in semiconductor manufacturing factory, *Application in Software and Computing*, **6**, 198-206.

Chen, T. (2006) A hybrid SOM-BPN approach to lot output time prediction in a wafer fab, *Neural Processing Letters*, **24**, 271-288.

Chen, T. (2007) Predicting Wafer-Lot Output Time With a Hybrid FCM-FBP Approach, *IEEE Transactions on Systems, Man, and Cybernetics- Part B: Cybernetics*, **37**(4), 784-793.

Chen, X., Fernandez, E. and Kelton, D. (2012) Optimization model selection for simulation-based approximate dynamic programming approaches in semiconductor manufacturing operations, *Winter Simulation Conference Proceedings*, 1977-1988.

Chung, Y., Kim, B., Park, S. and Seo, J. (2015) Reservation based dispatching rule for waferfab with engineering lots, *Proceedings of the 2015 Winter Simulation Conference*, 2974-2982.

Green, S.B. and Salkind, N.J. (2014) *Using SPSSfor windows and Macintosh: Analyzing and Understanding Data*, Seventh edition, Pearson, Upper Saddle River, NJ.

Hedman, R., Sundkvist, R., Almström, P. and Kinnander, A. (2013) Object-oriented Modeling Manufacturing Resources Using Work Study Inputs, *Procedia CIRP*, **7**(0), 443-448.

Hopp, W.J and Spearman, M.L. (2001) *Factory Physics: Foundations of Manufacturing Management*, Second edition, Irwin McGraw-Hill, New York, NY.

Kelton, D.W., Sadowski, R.P. and Sadowski, D.A. (2002) *Simulation with Arena*, Second edition, McGraw Hill, New York, NY.

Kempf, K. (1994) Detailed Description of Multiple Product Re-entrant Semiconductor Manufacturing System Example, *Prepared report*, Intel Corporation, Technology, and Manufacturing Group.

Kumar, S. and Nottestand, D.A. (2006) Capacity Design: an application using discrete-event simulation and designed experiments, *IIE Transactions*, **38**, 729-736.

Meidan, Y., Lerner B., Rabinowitz, G. and Hassoun, M. (2011) Cycle-Time Key Factor Identification and Prediction in Semiconductor Manufacturing Using Machine Learning and Data Mining, *IEEE Transactions on Semiconductor Manufacturing*, **24**(2).

Mittler, M., Gerlich, N. and Schömig, A.K. (1995) On Cycle Times and Interdeparture Times in Semiconductor Manufacturing, Technical Report No. 124, University of Würzburg, Germany.

Montevechi, J.A.B., Filho, R.G., Paiva, A.P., Costa, R.F.S. and Mederios, A.L. (2010) Sensitivity analysis in discrete-event simulation using fractional factorial designs, *Journal of Simulation*, **4**, 128-142.

Montgomery, D.C. (2014) *Design and analysis of engineering experiments*, Wiley, New York, NY.

Pearn, W.L., Tai, Y. and Lee, J.H. (2009) Statistical Approach for Cycle Time Estimation in Semiconductor Packaging Factories, *IEEE Transactions on Electronics Packaging Manufacturing*, **32**(3), 198-205.

Pinedo, M. (1995) *Scheduling: Theory, Algorithms, and Systems*, Prentice Hall, Englewood Cliffs, NJ.

Qi, C., Tang, T. K. and Sivakumar, A.I. (2002) Simulation based cause and effect analysis of cycle time and WIP in semiconductor wafer fabrication, *Proceedings of the Winter Simulation Conference.*

Rose, O. (1999) Estimation of the cycle time distribution of a wafer fab by a simple simulation model, *Proceedings of the 1999 International Conference on Semiconductor Manufacturing*, Operational Modeling and Simulation, 133- 138.

Shanthikumar, J.G., Ding, S. and Zhang, M.T. (2007) Queuing Theory for Semiconductor Manufacturing Systems: A Survey and Open Problems, *IEEE Transactions on Automation Science and Engineering*, **4**(4), 513- 522.

Sivakumar, A.I. and Chong, C.S. (2001) A simulation based analysis of cycle time distribution, and throughput in semiconductor backend manufacturing, *Computers in Industry*, **45**, 59-78.

Wang, C.N. and Wang, C.H. (2007) A simulated model for cycle time reduction by acquiring optimal lot size in semiconductor manufacturing, *The International Journal of Advanced Manufacturing Technology*, **34**, 1008-1015.

Yang, F., Ankenman, B.E., Nelson, B.L. (2008) Estimating Cycle Time Percentile Curves for Manufacturing Systems via Simulation, *INFORMS Journal on Computing,* **20**(4), 628-643.

APPENDIX A

“.MOD; AND “.EXP” FILES OF MINI-FAB SIMULATION MODEL

A. “.mod” and “.exp” files of mini-fab simulation model.

.mod file:

```
;
;
;     Model statements for module:  BasicProcess.Create 1 (Simulation Start)
;

63$           CREATE,        1,HoursToBaseTime(0.0),Type
A:HoursToBaseTime(EXPO(1)),1:NEXT(64$);

64$           ASSIGN:        Simulation Start.NumberOut=Simulation
Start.NumberOut + 1:NEXT(0$);


;
;
;     Model statements for module:  AdvancedProcess.ReadWrite 1 (Read TBA)
;
0$            READ,          MeanTBA:
                             ArrivalRate:NEXT(1$);


;
;
;     Model statements for module:  AdvancedProcess.ReadWrite 2 (Read MTBF)
;
1$            READ,          MeanTimeBetweenFailure:
                             MTBF:NEXT(2$);


;
;
;     Model statements for module:  AdvancedProcess.ReadWrite 3 (Read Unloading
Variance)
;
2$            READ,          Variance_Unloading:
                             UnloadVar:NEXT(3$);


;
;
;     Model statements for module:  AdvancedProcess.ReadWrite 4 (Read MTTR)
;
3$            READ,          RepairMeanTime:
                             MTTR:NEXT(4$);


;
;
;     Model statements for module:  BasicProcess.Assign 1 (Variables)
;
4$            ASSIGN:        LagTally=LagValue:
                             N_Exists=0:
                             BatchN=0:
                             BatchSize=0:
                             BatchCT=0:
```

```
                              DesiredTotal=5000:
                              LagValue=2000:
                              TruncationPoint=250000:
                              Machine3UpTime=EXPO(MTBF):
                              Machine3DownTime=GAMM(MTTR,0.25):NEXT(5$);


;
;
;     Model statements for module:  BasicProcess.Dispose 1 (Dispose Simulation
Start)
;
5$            ASSIGN:        Dispose Simulation Start.NumberOut=Dispose
Simulation Start.NumberOut + 1;
67$           DISPOSE:       Yes;


;
;
;     Model statements for module:  BasicProcess.Create 2 (Part X Arrival)
;

68$           CREATE,        1,MinutesToBaseTime(0.01),Part
X:MinutesToBaseTime(EXPO(1/ArrivalRate)):NEXT(69$);

69$           ASSIGN:        Part X Arrival.NumberOut=Part X Arrival.NumberOut
+ 1:NEXT(6$);


;
;
;     Model statements for module:  BasicProcess.Assign 2 (Part X Attributes)
;
6$            ASSIGN:        EntityType=1:
                              CreationTime=TNOW:
                              Entity.Sequence=PartFlow:NEXT(8$);


;
;
;     Model statements for module:  AdvancedTransfer.Station 1 (Order Release)
;

8$            STATION,       Order Release;
74$           DELAY:         0.0,,VA:NEXT(9$);


;
;
;     Model statements for module:  AdvancedTransfer.Route 1 (Route to Tool
Group 1)
;
9$            ROUTE:         0.,SEQ;


;
;
;     Model statements for module:  BasicProcess.Create 3 (Part Y Arrival)
;
```

```
75$           CREATE,        1,MinutesToBaseTime(0.01),Part
Y:MinutesToBaseTime(EXPO(1/0.003)):NEXT(76$);

76$           ASSIGN:        Part Y Arrival.NumberOut=Part Y Arrival.NumberOut
+ 1:NEXT(7$);


;
;
;     Model statements for module:  BasicProcess.Assign 3 (Paart Y Attribute)
;
7$            ASSIGN:        EntityType=2:
                             Entity.Sequence=PartFlow:NEXT(8$);


;
;
;     Model statements for module:  AdvancedTransfer.Station 2 (Station for
Tool Group 1)
;

10$           STATION,       TG 1;
81$           DELAY:         0.0,,VA:NEXT(11$);


;
;
;     Model statements for module:  BasicProcess.Decide 1 (Type of Batching)
;
11$           BRANCH,        1:
                             If,Entity.JobStep==1,82$,Yes:
                             Else,83$,Yes;
82$           ASSIGN:        Type of Batching.NumberOut True=Type of
Batching.NumberOut True + 1:NEXT(12$);

83$           ASSIGN:        Type of Batching.NumberOut False=Type of
Batching.NumberOut False + 1:NEXT(13$);


;
;
;     Model statements for module:  BasicProcess.Batch 1 (Batching Process 1)
;
12$           QUEUE,         Batching Process 1.Queue;
84$           GROUP,         ,Temporary:3,Last,Part X:NEXT(85$);

85$           ASSIGN:        Batching Process 1.NumberOut=Batching Process
1.NumberOut + 1:NEXT(16$);


;
;
;     Model statements for module:  BasicProcess.Process 1 (Loading for tool
group 1)
;
16$           ASSIGN:        Loading for tool group 1.NumberIn=Loading for tool
group 1.NumberIn + 1:
                             Loading for tool group 1.WIP=Loading for tool
group 1.WIP+1;
89$           QUEUE,         Loading for tool group 1.Queue;
```

```
88$           SEIZE,         2,NVA:
                             Machine 1,1:
                             Operator 1,1:NEXT(87$);

87$           DELAY:         NORM(20,2),,NVA;
134$          ASSIGN:        Loading for tool group 1.NumberOut=Loading for
tool group 1.NumberOut + 1:
                             Loading for tool group 1.WIP=Loading for tool
group 1.WIP-1:NEXT(17$);


;
;
;     Model statements for module:  BasicProcess.Process 2 (Release the
Operator 1)
;
17$           ASSIGN:        Release the Operator 1.NumberIn=Release the
Operator 1.NumberIn + 1:
                             Release the Operator 1.WIP=Release the Operator
1.WIP+1;
138$          DELAY:         0,,VA;
137$          RELEASE:       Operator 1,1;
185$          ASSIGN:        Release the Operator 1.NumberOut=Release the
Operator 1.NumberOut + 1:
                             Release the Operator 1.WIP=Release the Operator
1.WIP-1:NEXT(18$);


;
;
;     Model statements for module:  BasicProcess.Process 3 (Process for Tool
Group 1)
;
18$           ASSIGN:        Process for Tool Group 1.NumberIn=Process for Tool
Group 1.NumberIn + 1:
                             Process for Tool Group 1.WIP=Process for Tool
Group 1.WIP+1;
189$          DELAY:         ProcessingTime,,VA;
236$          ASSIGN:        Process for Tool Group 1.NumberOut=Process for
Tool Group 1.NumberOut + 1:
                             Process for Tool Group 1.WIP=Process for Tool
Group 1.WIP-1:NEXT(19$);


;
;
;     Model statements for module:  BasicProcess.Process 4 (Unloading for Tool
Group 1)
;
19$           ASSIGN:        Unloading for Tool Group 1.NumberIn=Unloading for
Tool Group 1.NumberIn + 1:
                             Unloading for Tool Group 1.WIP=Unloading for Tool
Group 1.WIP+1;
242$          QUEUE,         Unloading for Tool Group 1.Queue;
241$          SEIZE,         2,NVA:
                             Operator 1,1:NEXT(240$);

240$          DELAY:         NORM(40,40*UnloadVar),,NVA;
287$          ASSIGN:        Unloading for Tool Group 1.NumberOut=Unloading for
Tool Group 1.NumberOut + 1:
```

```
                             Unloading for Tool Group 1.WIP=Unloading for Tool
Group 1.WIP-1:NEXT(20$);


;
;
;     Model statements for module:  BasicProcess.Process 5 (Rlease Tool Group 1
Resources)
;
20$           ASSIGN:        Rlease Tool Group 1 Resources.NumberIn=Rlease Tool
Group 1 Resources.NumberIn + 1:
                             Rlease Tool Group 1 Resources.WIP=Rlease Tool
Group 1 Resources.WIP+1;
291$          DELAY:         0,,VA;
290$          RELEASE:       Machine 1,1:
                             Operator 1,1;
338$          ASSIGN:        Rlease Tool Group 1 Resources.NumberOut=Rlease
Tool Group 1 Resources.NumberOut + 1:
                             Rlease Tool Group 1 Resources.WIP=Rlease Tool
Group 1 Resources.WIP-1:NEXT(62$);


;
;
;     Model statements for module:  BasicProcess.Separate 1 (Unbatching for
Tool Group 1)
;
62$           SPLIT:         Entity.BatchN:NEXT(341$);

341$          ASSIGN:        Unbatching for Tool Group 1.NumberOut
Orig=Unbatching for Tool Group 1.NumberOut Orig + 1:NEXT(21$);


;
;
;     Model statements for module:  AdvancedTransfer.Route 2 (Route to Tool
Group 2)
;
21$           ROUTE:         0.,SEQ;


;
;
;     Model statements for module:  BasicProcess.Batch 2 (Batching Process 5)
;
13$           QUEUE,         Batching Process 5.Queue;
344$          GROUP,         EntityType,Temporary:3,Last,Part Y:NEXT(345$);

345$          ASSIGN:        Batching Process 5.NumberOut=Batching Process
5.NumberOut + 1:NEXT(14$);


;
;
;     Model statements for module:  BasicProcess.Assign 4 (Count Batches)
;
14$           ASSIGN:        BatchN=BatchN+1:NEXT(15$);


;
```

```
;
;     Model statements for module:  BasicProcess.Assign 5 (Batch Number)
;
15$           ASSIGN:        Entity.BatchN=BatchN:NEXT(16$);


;
;
;     Model statements for module:  AdvancedTransfer.Station 3 (Station for
Tool Group 2)
;

27$           STATION,       TG 2;
348$          DELAY:         0.0,,VA:NEXT(22$);


;
;
;     Model statements for module:  BasicProcess.Process 6 (Loading for Tool
Group 2)
;
22$           ASSIGN:        Loading for Tool Group 2.NumberIn=Loading for Tool
Group 2.NumberIn + 1:
                             Loading for Tool Group 2.WIP=Loading for Tool
Group 2.WIP+1;
352$          QUEUE,         Loading for Tool Group 2.Queue;
351$          SEIZE,         2,NVA:
                             Machine 2,1:
                             Operator 2,1:NEXT(350$);

350$          DELAY:         NORM(15,1.5),,NVA;
397$          ASSIGN:        Loading for Tool Group 2.NumberOut=Loading for
Tool Group 2.NumberOut + 1:
                             Loading for Tool Group 2.WIP=Loading for Tool
Group 2.WIP-1:NEXT(23$);


;
;
;     Model statements for module:  BasicProcess.Process 7 (Release Operator 2)
;
23$           ASSIGN:        Release Operator 2.NumberIn=Release Operator
2.NumberIn + 1:
                             Release Operator 2.WIP=Release Operator 2.WIP+1;
401$          DELAY:         0,,VA;
400$          RELEASE:       Operator 2,1;
448$          ASSIGN:        Release Operator 2.NumberOut=Release Operator
2.NumberOut + 1:
                             Release Operator 2.WIP=Release Operator 2.WIP-
1:NEXT(24$);


;
;
;     Model statements for module:  BasicProcess.Process 8 (Process for Tool
Group 2)
;
24$           ASSIGN:        Process for Tool Group 2.NumberIn=Process for Tool
Group 2.NumberIn + 1:
```

```
                              Process for Tool Group 2.WIP=Process for Tool
Group 2.WIP+1;
452$          DELAY:          ProcessingTime,,VA;
499$          ASSIGN:         Process for Tool Group 2.NumberOut=Process for
Tool Group 2.NumberOut + 1:
                              Process for Tool Group 2.WIP=Process for Tool
Group 2.WIP-1:NEXT(29$);


;
;
;     Model statements for module:  BasicProcess.Decide 2 (pass or fail)
;
29$           BRANCH,         1:
                              With,(2)/100,502$,Yes:
                              Else,503$,Yes;
502$          ASSIGN:         pass or fail.NumberOut True=pass or fail.NumberOut
True + 1:NEXT(30$);

503$          ASSIGN:         pass or fail.NumberOut False=pass or
fail.NumberOut False + 1:NEXT(25$);


;
;
;     Model statements for module:  BasicProcess.Process 11 (Rework)
;
30$           ASSIGN:         Rework.NumberIn=Rework.NumberIn + 1:
                              Rework.WIP=Rework.WIP+1;
507$          QUEUE,          Rework.Queue;
506$          SEIZE,          2,VA:
                              Machine R,1:
                              Operator R,1:NEXT(505$);

505$          DELAY:          ProcessingTime/2,,VA;
504$          RELEASE:        Machine R,1:
                              Operator R,1;
552$          ASSIGN:         Rework.NumberOut=Rework.NumberOut + 1:
                              Rework.WIP=Rework.WIP-1:NEXT(25$);


;
;
;     Model statements for module:  BasicProcess.Process 9 (Unloading for Tool
Group 2)
;
25$           ASSIGN:         Unloading for Tool Group 2.NumberIn=Unloading for
Tool Group 2.NumberIn + 1:
                              Unloading for Tool Group 2.WIP=Unloading for Tool
Group 2.WIP+1;
558$          QUEUE,          Unloading for Tool Group 2.Queue;
557$          SEIZE,          2,VA:
                              Operator 2,1:NEXT(556$);

556$          DELAY:          NORM(15,15*UnloadVar),,VA;
603$          ASSIGN:         Unloading for Tool Group 2.NumberOut=Unloading for
Tool Group 2.NumberOut + 1:
                              Unloading for Tool Group 2.WIP=Unloading for Tool
Group 2.WIP-1:NEXT(26$);
```

```
;
;
;     Model statements for module:  BasicProcess.Process 10 (Release Tool Group
2 Resources)
;
26$           ASSIGN:        Release Tool Group 2 Resources.NumberIn=Release
Tool Group 2 Resources.NumberIn + 1:
                             Release Tool Group 2 Resources.WIP=Release Tool
Group 2 Resources.WIP+1;
607$          DELAY:         0,,VA;
606$          RELEASE:       Machine 2,1:
                             Operator 2,1;
654$          ASSIGN:        Release Tool Group 2 Resources.NumberOut=Release
Tool Group 2 Resources.NumberOut + 1:
                             Release Tool Group 2 Resources.WIP=Release Tool
Group 2 Resources.WIP-1:NEXT(28$);


;
;
;     Model statements for module:  AdvancedTransfer.Route 3 (Route to Tool
Group 3)
;
28$           ROUTE:         0.,SEQ;


;
;
;     Model statements for module:  AdvancedTransfer.Station 5 (Station for
Tool Group 3)
;

36$           STATION,       TG 3;
659$          DELAY:         0.0,,VA:NEXT(38$);


;
;
;     Model statements for module:  BasicProcess.Decide 4 (Setup?)
;
38$           BRANCH,        1:
                             If,PartsSetup == EntityType && Step ==
Entity.JobStep,39$,Yes:
                             If,PartsSetup==EntityType,40$,Yes:
                             If,Step==Entity.JobStep,41$,Yes:
                             Else,42$,Yes;

;
;
;     Model statements for module:  BasicProcess.Assign 13 (Different Step
Different Type)
;
42$           ASSIGN:        Step=Entity.JobStep:
                             PartsSetup=Entity.Type:
                             SetupTime=NORM(12,6):NEXT(31$);


;
;
```

```
;     Model statements for module:  BasicProcess.Process 17 (SetUp and Loading
for Tool Group 3)
;
31$           ASSIGN:        SetUp and Loading for Tool Group 3.NumberIn=SetUp
and Loading for Tool Group 3.NumberIn + 1:
                             SetUp and Loading for Tool Group 3.WIP=SetUp and
Loading for Tool Group 3.WIP+1;
665$          QUEUE,         SetUp and Loading for Tool Group 3.Queue;
664$          SEIZE,         2,VA:
                             Machine 3,1:
                             Operator 3,1:NEXT(663$);

663$          DELAY:         SetupTime+NORM(10,1),,VA;
710$          ASSIGN:        SetUp and Loading for Tool Group 3.NumberOut=SetUp
and Loading for Tool Group 3.NumberOut + 1:
                             SetUp and Loading for Tool Group 3.WIP=SetUp and
Loading for Tool Group 3.WIP-1:NEXT(32$);


;
;
;     Model statements for module:  BasicProcess.Process 18 (Release Operator
3)
;
32$           ASSIGN:        Release Operator 3.NumberIn=Release Operator
3.NumberIn + 1:
                             Release Operator 3.WIP=Release Operator 3.WIP+1;
714$          DELAY:         0,,VA;
713$          RELEASE:       Operator 3,1;
761$          ASSIGN:        Release Operator 3.NumberOut=Release Operator
3.NumberOut + 1:
                             Release Operator 3.WIP=Release Operator 3.WIP-
1:NEXT(33$);


;
;
;     Model statements for module:  BasicProcess.Process 19 (Process for Tool
Group 3)
;
33$           ASSIGN:        Process for Tool Group 3.NumberIn=Process for Tool
Group 3.NumberIn + 1:
                             Process for Tool Group 3.WIP=Process for Tool
Group 3.WIP+1;
765$          DELAY:         ProcessingTime,,VA;
812$          ASSIGN:        Process for Tool Group 3.NumberOut=Process for
Tool Group 3.NumberOut + 1:
                             Process for Tool Group 3.WIP=Process for Tool
Group 3.WIP-1:NEXT(34$);


;
;
;     Model statements for module:  BasicProcess.Process 20 (Unloading for Tool
Group 3)
;
34$           ASSIGN:        Unloading for Tool Group 3.NumberIn=Unloading for
Tool Group 3.NumberIn + 1:
                             Unloading for Tool Group 3.WIP=Unloading for Tool
Group 3.WIP+1;
```

```
818$          QUEUE,         Unloading for Tool Group 3.Queue;
817$          SEIZE,         2,NVA:
                             Operator 3,1:NEXT(816$);

816$          DELAY:         NORM(10,UnloadVar*10),,NVA;
863$          ASSIGN:        Unloading for Tool Group 3.NumberOut=Unloading for
Tool Group 3.NumberOut + 1:
                             Unloading for Tool Group 3.WIP=Unloading for Tool
Group 3.WIP-1:NEXT(35$);


;
;
;     Model statements for module:  BasicProcess.Process 21 (Release Tool Group
3 Resources)
;
35$           ASSIGN:        Release Tool Group 3 Resources.NumberIn=Release
Tool Group 3 Resources.NumberIn + 1:
                             Release Tool Group 3 Resources.WIP=Release Tool
Group 3 Resources.WIP+1;
867$          DELAY:         0,,VA;
866$          RELEASE:       Machine 3,1:
                             Operator 3,1;
914$          ASSIGN:        Release Tool Group 3 Resources.NumberOut=Release
Tool Group 3 Resources.NumberOut + 1:
                             Release Tool Group 3 Resources.WIP=Release Tool
Group 3 Resources.WIP-1:NEXT(37$);


;
;
;     Model statements for module:  AdvancedTransfer.Route 5 (Route to Collect
Statistics)
;
37$           ROUTE:         0.,SEQ;


;
;
;     Model statements for module:  BasicProcess.Assign 10 (Same Step Same
Type)
;
39$           ASSIGN:        Step=Entity.JobStep:
                             PartsSetup=EntityType:
                             SetupTime=0:NEXT(31$);


;
;
;     Model statements for module:  BasicProcess.Assign 11 (Different Step Same
Type)
;
40$           ASSIGN:        Step=Entity.JobStep:
                             PartsSetup=Entity.Type:
                             SetupTime=NORM(10,5):NEXT(31$);


;
;
```

```
;     Model statements for module:  BasicProcess.Assign 12 (Same Step Different
Type)
;
41$           ASSIGN:        Step=Entity.JobStep:
                             PartsSetup=Entity.Type:
                             SetupTime=NORM(5,2.5):NEXT(31$);


;
;
;     Model statements for module:  AdvancedTransfer.Station 6 (Exit System
Station)
;

44$           STATION,       Exit System;
919$          DELAY:         0.0,,VA:NEXT(45$);


;
;
;     Model statements for module:  BasicProcess.Decide 6 (Part Type X?)
;
45$           BRANCH,        1:
                             If,Entity.Type==Part X,920$,Yes:
                             Else,921$,Yes;
920$          ASSIGN:        Part Type X?.NumberOut True=Part Type X?.NumberOut
True + 1:NEXT(54$);

921$          ASSIGN:        Part Type X?.NumberOut False=Part Type
X?.NumberOut False + 1:NEXT(51$);


;
;
;     Model statements for module:  BasicProcess.Decide 8 (At Truncation
Point?)
;
54$           BRANCH,        1:
                             If,TNOW<TruncationPoint,922$,Yes:
                             Else,923$,Yes;
922$          ASSIGN:        At Truncation Point?.NumberOut True=At Truncation
Point?.NumberOut True + 1:NEXT(46$);

923$          ASSIGN:        At Truncation Point?.NumberOut False=At Truncation
Point?.NumberOut False + 1:NEXT(59$);


;
;
;     Model statements for module:  BasicProcess.Dispose 3 (Dispose
PreTruncation Values)
;
46$           ASSIGN:        Dispose PreTruncation Values.NumberOut=Dispose
PreTruncation Values.NumberOut + 1;
924$          DISPOSE:       Yes;


;
;
```

```
;     Model statements for module:  BasicProcess.Assign 23 (Cycle Time
Calculation X)
;
59$           ASSIGN:        CycleTime=TNOW - CreationTime:NEXT(55$);


;
;
;     Model statements for module:  BasicProcess.Assign 20 (Update batch CT)
;
55$           ASSIGN:        BatchCT=BatchCT + CycleTime:NEXT(58$);


;
;
;     Model statements for module:  BasicProcess.Assign 22 (Update batch size)
;
58$           ASSIGN:        BatchSize=BatchSize+1:NEXT(56$);


;
;
;     Model statements for module:  BasicProcess.Decide 9 (Is Batch Complete?)
;
56$           BRANCH,        1:
                             If,BatchSize==3,925$,Yes:
                             Else,926$,Yes;
925$          ASSIGN:        Is Batch Complete?.NumberOut True=Is Batch
Complete?.NumberOut True + 1:NEXT(57$);

926$          ASSIGN:        Is Batch Complete?.NumberOut False=Is Batch
Complete?.NumberOut False + 1:NEXT(60$);


;
;
;     Model statements for module:  BasicProcess.Assign 21 (Average batch CT)
;
57$           ASSIGN:        BatchCT=BatchCT/3:NEXT(61$);


;
;
;     Model statements for module:  BasicProcess.Assign 24 (Assign Batch CT to
Attribute)
;
61$           ASSIGN:        BatchedCT=BatchCT:NEXT(48$);


;
;
;     Model statements for module:  BasicProcess.Assign 18 (Increment LagTally
and Reset Batch Variables)
;
48$           ASSIGN:        LagTally=LagTally+1:
                             BatchCT=0:
                             BatchSize=0:NEXT(47$);


;
```

```
;
;     Model statements for module:  BasicProcess.Decide 7 (Long Enough Lag X?)
;
47$           BRANCH,        1:
                             If,LagTally>LagValue,927$,Yes:
                             Else,928$,Yes;
927$          ASSIGN:        Long Enough Lag X?.NumberOut True=Long Enough Lag
X?.NumberOut True + 1:NEXT(50$);

928$          ASSIGN:        Long Enough Lag X?.NumberOut False=Long Enough Lag
X?.NumberOut False + 1:NEXT(49$);


;
;
;     Model statements for module:  BasicProcess.Assign 19 (Reset_Lag_X)
;
50$           ASSIGN:        LagTally=0:NEXT(52$);


;
;
;     Model statements for module:  BasicProcess.Record 1 (Number of  Exits)
;
52$           COUNT:         Exits,1:NEXT(53$);


;
;
;     Model statements for module:  AdvancedProcess.ReadWrite 5 (Export Product
X Data)
;
53$           WRITE,         XData:
                             BatchedCT:NEXT(43$);


;
;
;     Model statements for module:  BasicProcess.Dispose 2 (Dispose)
;
43$           ASSIGN:        Dispose.NumberOut=Dispose.NumberOut + 1;
929$          DISPOSE:       Yes;


;
;
;     Model statements for module:  BasicProcess.Dispose 4 (Dispose Between
Lags)
;
49$           ASSIGN:        Dispose Between Lags.NumberOut=Dispose Between
Lags.NumberOut + 1;
930$          DISPOSE:       Yes;


;
;
;     Model statements for module:  BasicProcess.Dispose 6 (Dispose Between
Complete Batches)
;
```

```
60$           ASSIGN:        Dispose Between Complete Batches.NumberOut=Dispose
Between Complete Batches.NumberOut + 1;
931$          DISPOSE:       Yes;


;
;
;     Model statements for module:  BasicProcess.Dispose 5 (Dispose Part Y)
;
51$           ASSIGN:        Dispose Part Y.NumberOut=Dispose Part Y.NumberOut
+ 1;
932$          DISPOSE:       Yes;
```

.exp file:

```
PROJECT,      "Unnamed
Project","tanushree.salvi@gmail.com",,,No,Yes,Yes,Yes,No,No,No,No,No,No;

ATTRIBUTES:   EntityType,DATATYPE(Real):
              Entity.BatchN,DATATYPE(Real):
              ProcessingTime,DATATYPE(Real):
              SetupTime,DATATYPE(Real):
              BatchedCT,DATATYPE(Real):
              CreationTime,DATATYPE(Real):
              CycleTime,DATATYPE(Real);

FILES:        XData,"T:\Thesis\Model\XData.txt",Sequential,Free
Format,Dispose,,Hold:

MeanTimeBetweenFailure,"T:\Thesis\Model\MeanTimeBetweenFailure.txt",Sequential,
Free Format,Dispose,,Hold:

RepairMeanTime,"T:\Thesis\Model\RepairMeanTime.txt",Sequential,Free
Format,Dispose,,Hold:
              MeanTBA,"T:\Thesis\Model\MeanTBA.txt",Sequential,Free
Format,Dispose,,Hold:

Variance_Unloading,"T:\Thesis\Model\Variance_Unloading.txt",Sequential,Free
Format,Dispose,,Hold;

VARIABLES:    At Truncation Point?.NumberOut
False,CLEAR(Statistics),CATEGORY("Exclude"):
              Machine3UpTime,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
              Release Operator
3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              Rlease Tool Group 1
Resources.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
              Rework.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              Release the Operator
1.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              Process for Tool Group
3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              Loading for tool group 1.WIP,CLEAR(System),CATEGORY("Exclude-
Exclude"),DATATYPE(Real):
              UnloadVar,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
              Dispose PreTruncation
Values.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              SetUp and Loading for Tool Group
3.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
              MTBF,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
              Unloading for Tool Group
1.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              Release Operator 2.WIP,CLEAR(System),CATEGORY("Exclude-
Exclude"),DATATYPE(Real):
              Type of Batching.NumberOut
False,CLEAR(Statistics),CATEGORY("Exclude"):
              Dispose Simulation
Start.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              pass or fail.NumberOut
True,CLEAR(Statistics),CATEGORY("Exclude"):
```

```
              Unloading for Tool Group
2.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
              Dispose.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              At Truncation Point?.NumberOut
True,CLEAR(Statistics),CATEGORY("Exclude"):
              Is Batch Complete?.NumberOut
False,CLEAR(Statistics),CATEGORY("Exclude"):
              Process for Tool Group
1.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
              Process for Tool Group
3.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
              Release Tool Group 3
Resources.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              Release Tool Group 3
Resources.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
              Loading for Tool Group
2.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
              Dispose Between Complete
Batches.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              SetUp and Loading for Tool Group
3.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"),DATATYPE(Real):
              Unloading for Tool Group 2.WIP,CLEAR(System),CATEGORY("Exclude-
Exclude"),DATATYPE(Real):
              Release Operator
3.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
              Process for Tool Group 1.WIP,CLEAR(System),CATEGORY("Exclude-
Exclude"),DATATYPE(Real):
              Loading for Tool Group
2.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              Part Y Arrival.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              BatchSize,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
              Release Operator
2.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              MTTR,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
              Process for Tool Group
2.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              LagValue,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
              Batching Process
5.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              Unloading for Tool Group
3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              Part Type X?.NumberOut
False,CLEAR(Statistics),CATEGORY("Exclude"):
              Is Batch Complete?.NumberOut
True,CLEAR(Statistics),CATEGORY("Exclude"):
              Machine3DownTime,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
              BatchCT,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
              Step,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
              N_Exists,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
              Process for Tool Group 3.WIP,CLEAR(System),CATEGORY("Exclude-
Exclude"),DATATYPE(Real):
              Rework.WIP,CLEAR(System),CATEGORY("Exclude-
Exclude"),DATATYPE(Real):
```

```
              ArrivalRate,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
              Release Tool Group 2
Resources.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
              Release Tool Group 2
Resources.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"),DATATYPE(Real):
              SetUp and Loading for Tool Group
3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              Release the Operator
1.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
              Rlease Tool Group 1
Resources.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              Type of Batching.NumberOut
True,CLEAR(Statistics),CATEGORY("Exclude"):
              Loading for tool group
1.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              pass or fail.NumberOut
False,CLEAR(Statistics),CATEGORY("Exclude"):
              BatchN,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
              Rlease Tool Group 1
Resources.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"),DATATYPE(Real):
              Release the Operator 1.WIP,CLEAR(System),CATEGORY("Exclude-
Exclude"),DATATYPE(Real):
              Simulation Start.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              Unloading for Tool Group
1.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
              Unloading for Tool Group
3.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
              LagTally,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
              Long Enough Lag X?.NumberOut
False,CLEAR(Statistics),CATEGORY("Exclude"):
              Unloading for Tool Group 1.WIP,CLEAR(System),CATEGORY("Exclude-
Exclude"),DATATYPE(Real):
              Loading for Tool Group 2.WIP,CLEAR(System),CATEGORY("Exclude-
Exclude"),DATATYPE(Real):
              Process for Tool Group
2.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
              Release Operator 3.WIP,CLEAR(System),CATEGORY("Exclude-
Exclude"),DATATYPE(Real):
              Unloading for Tool Group
2.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              Loading for tool group
1.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
              Rework.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
              Process for Tool Group
1.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              Release Operator
2.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
              PartsSetup,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
              Release Tool Group 3
Resources.WIP,CLEAR(System),CATEGORY("Exclude-Exclude"),DATATYPE(Real):
              Batching Process
1.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              Long Enough Lag X?.NumberOut
True,CLEAR(Statistics),CATEGORY("Exclude"):
              TruncationPoint,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
```

```
                DesiredTotal,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
                Dispose Between
Lags.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
                Part Type X?.NumberOut
True,CLEAR(Statistics),CATEGORY("Exclude"):
                Unloading for Tool Group 3.WIP,CLEAR(System),CATEGORY("Exclude-
Exclude"),DATATYPE(Real):
                Process for Tool Group 2.WIP,CLEAR(System),CATEGORY("Exclude-
Exclude"),DATATYPE(Real):
                Release Tool Group 2
Resources.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
                Dispose Part Y.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
                Unbatching for Tool Group 1.NumberOut
Orig,CLEAR(Statistics),CATEGORY("Exclude"):
                Part X Arrival.NumberOut,CLEAR(Statistics),CATEGORY("Exclude");

QUEUES:         Unloading for Tool Group 1.Queue,FIFO,,AUTOSTATS(Yes,,):
                Batching Process 5.Queue,FIFO,,AUTOSTATS(Yes,,):
                Unloading for Tool Group 2.Queue,FIFO,,AUTOSTATS(Yes,,):
                Batching Process 1.Queue,FIFO,,AUTOSTATS(Yes,,):
                Loading for tool group 1.Queue,FIFO,,AUTOSTATS(Yes,,):
                Rework.Queue,FIFO,,AUTOSTATS(Yes,,):
                Loading for Tool Group 2.Queue,FIFO,,AUTOSTATS(Yes,,):
                Unloading for Tool Group 3.Queue,FIFO,,AUTOSTATS(Yes,,):
                SetUp and Loading for Tool Group 3.Queue,FIFO,,AUTOSTATS(Yes,,);

PICTURES:       Picture.Airplane:
                Picture.Green Ball:
                Picture.Blue Page:
                Picture.Telephone:
                Picture.Blue Ball:
                Picture.Yellow Page:
                Picture.EMail:
                Picture.Yellow Ball:
                Picture.Bike:
                Picture.Report:
                Picture.Van:
                Picture.Widgets:
                Picture.Envelope:
                Picture.Fax:
                Picture.Truck:
                Picture.Person:
                Picture.Letter:
                Picture.Box:
                Picture.Woman:
                Picture.Package:
                Picture.Man:
                Picture.Diskette:
                Picture.Boat:
                Picture.Red Page:
                Picture.Ball:
                Picture.Green Page:
                Picture.Red Ball;

FAILURES:       PM,Time(10080.000000000000000,60.000000000000000,):
                Condition
Checks,Time(43200.000000000000000,360.000000000000000,):
                EM,Time(DaysToBaseTime(Machine3UpTime),Machine3DownTime,);
```

```
RESOURCES:    Operator
1,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes,,):
              Operator
2,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes,,):
              Operator
3,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes,,):
              Operator
R,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes,,):
              Machine
1,Capacity(2),,,COST(0.0,0.0,0.0),CATEGORY(Resources),FAILURE(Condition
Checks,Wait),FAILURE(PM,Wait),
              AUTOSTATS(Yes,,):
              Machine
2,Capacity(2),,,COST(0.0,0.0,0.0),CATEGORY(Resources),FAILURE(Condition
Checks,Wait),FAILURE(PM,Wait),
              AUTOSTATS(Yes,,):
              Machine
3,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),FAILURE(Condition
Checks,Wait),FAILURE(PM,Wait),
              FAILURE(EM,Preempt),AUTOSTATS(Yes,,):
              Machine
R,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),FAILURE(Condition
Checks,Wait),FAILURE(PM,Wait),
              AUTOSTATS(Yes,,);

STATIONS:     Order Release,,,Order Release,AUTOSTATS(Yes,,):
              Exit System,,,Exit System,AUTOSTATS(Yes,,):
              TG 1,,,TG 1,AUTOSTATS(Yes,,):
              TG 2,,,TG 2,AUTOSTATS(Yes,,):
              TG 3,,,TG 3,AUTOSTATS(Yes,,);

SEQUENCES:    PartFLow,TG 1,STEPNAME=Process
1,,,ProcessingTime=NORM(292.5,11.25)&TG 2,STEPNAME=Process 2,,,ProcessingTime=
              NORM(39,1.5)&TG 3,STEPNAME=Process
3,,,ProcessingTime=NORM(71.5,2.75)&TG 2,STEPNAME=Process 4,,,ProcessingTime=
              NORM(65,2.5)&TG 1,STEPNAME=Process
5,,,ProcessingTime=NORM(331.5,12.75)&TG 3,STEPNAME=Process 6,,,ProcessingTime=
              NORM(13,0.5)&Exit System,STEPNAME=Process 7;

COUNTERS:     Exits,,,,DATABASE(,"Count","User Specified","Exits");

REPLICATE,    1,,,Yes,Yes,,NC(Exits)>DesiredTotal,,24,Minutes,No,No,,,Yes,No;

ENTITIES:     Part X,Picture.Report,0.0,0.0,0.0,0.0,0.0,0.0,AUTOSTATS(Yes,,):
              Part Y,Picture.Report,0.0,0.0,0.0,0.0,0.0,0.0,AUTOSTATS(Yes,,):
              Type A,Picture.Report,0.0,0.0,0.0,0.0,0.0,0.0,AUTOSTATS(Yes,,);

ACTIVITYAREAS: Order Release,0,,AUTOSTATS(Yes,,):
              Exit System,0,,AUTOSTATS(Yes,,):
              TG 1,0,,AUTOSTATS(Yes,,):
              TG 2,0,,AUTOSTATS(Yes,,):
              TG 3,0,,AUTOSTATS(Yes,,);
```

APPENDIX B

“.MOD” AND “. EXP” FILE OF JOB SHOP MODEL

B. ".mod" and ".exp" files of mini-fab simulation model.

.mod file:

```
;
;
;     Model statements for module:  BasicProcess.Create 1 (Part X Arrival)
;

48$           CREATE,        1,MinutesToBaseTime(0.01),Entity
1:MinutesToBaseTime(EXPO(ArrivalRate)):NEXT(49$);

49$           ASSIGN:        Part X Arrival.NumberOut=Part X Arrival.NumberOut
+ 1:NEXT(0$);


;
;
;     Model statements for module:  BasicProcess.Assign 1 (Attributes for Part
X)
;
0$            ASSIGN:        ArrivalTime=TNOW:
                             Type=1:
                             Entity.Sequence=Sequence for Part X:NEXT(3$);


;
;
;     Model statements for module:  AdvancedTransfer.Route 1 (Route to Tool
Groups)
;
3$            ROUTE:         0,SEQ;


;
;
;     Model statements for module:  BasicProcess.Create 2 (Part Y Arrival)
;

52$           CREATE,        1,MinutesToBaseTime(0.01),Entity
1:MinutesToBaseTime(EXPO(ArrivalRate_Y)):NEXT(53$);

53$           ASSIGN:        Part Y Arrival.NumberOut=Part Y Arrival.NumberOut
+ 1:NEXT(1$);


;
;
;     Model statements for module:  BasicProcess.Assign 2 (Attributes for Part
Y)
;
1$            ASSIGN:        ArrivalTime=TNOW:
                             Type=2:
                             Entity.Sequence=Sequence for Part Y:NEXT(3$);
```

```
;
;
;     Model statements for module:  BasicProcess.Create 3 (Part Z Arrival)
;

56$           CREATE,        1,MinutesToBaseTime(0.01),Entity
1:MinutesToBaseTime(EXPO(ArrivalRate_Z)):NEXT(57$);

57$           ASSIGN:        Part Z Arrival.NumberOut=Part Z Arrival.NumberOut
+ 1:NEXT(2$);


;
;
;     Model statements for module:  BasicProcess.Assign 3 (Attributes for Part
Z)
;
2$            ASSIGN:        Type=3:
                             ArrivalTime=TNOW:
                             Entity.Sequence=Sequence for Part Z:NEXT(3$);


;
;
;     Model statements for module:  AdvancedTransfer.Station 1 (Station for
Tool Group 1)
;

8$            STATION,       TG 1;
62$           DELAY:         0.0,,VA:NEXT(4$);


;
;
;     Model statements for module:  BasicProcess.Process 1 (Tool Group 1)
;
4$            ASSIGN:        Tool Group 1.NumberIn=Tool Group 1.NumberIn + 1:
                             Tool Group 1.WIP=Tool Group 1.WIP+1;
66$           QUEUE,         Tool Group 1.Queue;
65$           SEIZE,         2,VA:
                             Machine 1,1:
                             Operator 1,1:NEXT(64$);

64$           DELAY:         ProcessTime,,VA;
63$           RELEASE:       Machine 1,1:
                             Operator 1,1;
111$          ASSIGN:        Tool Group 1.NumberOut=Tool Group 1.NumberOut + 1:
                             Tool Group 1.WIP=Tool Group 1.WIP-1:NEXT(12$);


;
;
;     Model statements for module:  AdvancedTransfer.Route 2 (Route for Tool
Group 1)
;
12$           ROUTE:         0.,SEQ;


;
```

```
;
;     Model statements for module:  AdvancedTransfer.Station 2 (Station for
Tool Group 2)
;

9$            STATION,       TG 2;
116$          DELAY:         0.0,,VA:NEXT(5$);


;
;
;     Model statements for module:  BasicProcess.Process 2 (Tool Group 2)
;
5$            ASSIGN:        Tool Group 2.NumberIn=Tool Group 2.NumberIn + 1:
                             Tool Group 2.WIP=Tool Group 2.WIP+1;
120$          QUEUE,         Tool Group 2.Queue;
119$          SEIZE,         2,VA:
                             Machine 2,1:
                             Operator 2,1:NEXT(118$);

118$          DELAY:         ProcessTime,,VA;
117$          RELEASE:       Machine 2,1:
                             Operator 2,1;
165$          ASSIGN:        Tool Group 2.NumberOut=Tool Group 2.NumberOut + 1:
                             Tool Group 2.WIP=Tool Group 2.WIP-1:NEXT(13$);


;
;
;     Model statements for module:  AdvancedTransfer.Route 3 (Route for Tool
Group 2)
;
13$           ROUTE:         0.,SEQ;


;
;
;     Model statements for module:  AdvancedTransfer.Station 3 (Station for
Tool Group 3)
;

10$           STATION,       TG 3;
170$          DELAY:         0.0,,VA:NEXT(6$);


;
;
;     Model statements for module:  BasicProcess.Process 3 (Tool Group 3)
;
6$            ASSIGN:        Tool Group 3.NumberIn=Tool Group 3.NumberIn + 1:
                             Tool Group 3.WIP=Tool Group 3.WIP+1;
174$          QUEUE,         Tool Group 3.Queue;
173$          SEIZE,         2,VA:
                             Machine 3,1:
                             Operator 3,1:NEXT(172$);

172$          DELAY:         ProcessTime,,VA;
171$          RELEASE:       Machine 3,1:
                             Operator 3,1;
219$          ASSIGN:        Tool Group 3.NumberOut=Tool Group 3.NumberOut + 1:
```

```
                            Tool Group 3.WIP=Tool Group 3.WIP-1:NEXT(14$);


;
;
;     Model statements for module:  AdvancedTransfer.Route 4 (Route for Tool
Group 3)
;
14$           ROUTE:         0.,SEQ;


;
;
;     Model statements for module:  AdvancedTransfer.Station 4 (Station for
Tool Group 4)
;

11$           STATION,       TG 4;
224$          DELAY:         0.0,,VA:NEXT(7$);


;
;
;     Model statements for module:  BasicProcess.Process 4 (Tool Group 4)
;
7$            ASSIGN:        Tool Group 4.NumberIn=Tool Group 4.NumberIn + 1:
                             Tool Group 4.WIP=Tool Group 4.WIP+1;
228$          QUEUE,         Tool Group 4.Queue;
227$          SEIZE,         2,VA:
                             Machine 4,1:
                             Operator 4,1:NEXT(226$);

226$          DELAY:         ProcessTime,,VA;
225$          RELEASE:       Machine 4,1:
                             Operator 4,1;
273$          ASSIGN:        Tool Group 4.NumberOut=Tool Group 4.NumberOut + 1:
                             Tool Group 4.WIP=Tool Group 4.WIP-1:NEXT(15$);


;
;
;     Model statements for module:  AdvancedTransfer.Route 5 (Route for Tool
Group 4)
;
15$           ROUTE:         0.,SEQ;


;
;
;     Model statements for module:  AdvancedTransfer.Station 5 (System Exit)
;

16$           STATION,       Collect Stats;
278$          DELAY:         0.0,,VA:NEXT(21$);


;
;
;     Model statements for module:  BasicProcess.Decide 3 (At Truncation Point)
;
```

```
21$           BRANCH,        1:
                             If,TNOW<TruncationPoint,279$,Yes:
                             Else,280$,Yes;
279$          ASSIGN:        At Truncation Point.NumberOut True=At Truncation
Point.NumberOut True + 1:NEXT(22$);

280$          ASSIGN:        At Truncation Point.NumberOut False=At Truncation
Point.NumberOut False + 1:NEXT(23$);


;
;
;     Model statements for module:  BasicProcess.Dispose 3 (Dispose rest)
;
22$           ASSIGN:        Dispose rest.NumberOut=Dispose rest.NumberOut + 1;
281$          DISPOSE:       Yes;


;
;
;     Model statements for module:  BasicProcess.Assign 8 (Increament Lag_X)
;
23$           ASSIGN:        LagX=LagX+1:NEXT(24$);


;
;
;     Model statements for module:  BasicProcess.Decide 5 (Lag Long enough?)
;
24$           BRANCH,        1:
                             If,LagX>LagValue,282$,Yes:
                             Else,283$,Yes;
282$          ASSIGN:        Lag Long enough?.NumberOut True=Lag Long
enough?.NumberOut True + 1:NEXT(26$);

283$          ASSIGN:        Lag Long enough?.NumberOut False=Lag Long
enough?.NumberOut False + 1:NEXT(25$);


;
;
;     Model statements for module:  BasicProcess.Assign 9 (Reset Lag)
;
26$           ASSIGN:        CT=TNOW- ArrivalTime:
                             LagX=0:NEXT(17$);


;
;
;     Model statements for module:  BasicProcess.Record 1 (Exists)
;
17$           COUNT:         Exists,1:NEXT(31$);


;
;
;     Model statements for module:  BasicProcess.Decide 8 (Number Replication
1)
;
31$           BRANCH,        1:
```

```
                              If,RepNo==1,284$,Yes:
                              Else,285$,Yes;
284$          ASSIGN:         Number Replication 1.NumberOut True=Number
Replication 1.NumberOut True + 1:NEXT(27$);

285$          ASSIGN:         Number Replication 1.NumberOut False=Number
Replication 1.NumberOut False + 1:NEXT(32$);


;
;
;     Model statements for module:  AdvancedProcess.ReadWrite 2 (Export Data1)
;
27$           WRITE,          Output:
                              CT:NEXT(28$);


;
;
;     Model statements for module:  BasicProcess.Dispose 5 (Dispose 5)
;
28$           ASSIGN:         Dispose 5.NumberOut=Dispose 5.NumberOut + 1;
286$          DISPOSE:        Yes;


;
;
;     Model statements for module:  BasicProcess.Decide 9 (Number Replication
2)
;
32$           BRANCH,         1:
                              If,RepNo==2,287$,Yes:
                              Else,288$,Yes;
287$          ASSIGN:         Number Replication 2.NumberOut True=Number
Replication 2.NumberOut True + 1:NEXT(33$);

288$          ASSIGN:         Number Replication 2.NumberOut False=Number
Replication 2.NumberOut False + 1:NEXT(34$);


;
;
;     Model statements for module:  AdvancedProcess.ReadWrite 6 (Export Data2)
;
33$           WRITE,          Output:
                              CT:NEXT(28$);


;
;
;     Model statements for module:  BasicProcess.Decide 10 (Number Replication
3)
;
34$           BRANCH,         1:
                              If,RepNo==3,289$,Yes:
                              Else,290$,Yes;
289$          ASSIGN:         Number Replication 3.NumberOut True=Number
Replication 3.NumberOut True + 1:NEXT(37$);
```

```
290$          ASSIGN:        Number Replication 3.NumberOut False=Number
Replication 3.NumberOut False + 1:NEXT(35$);


;
;
;     Model statements for module:  AdvancedProcess.ReadWrite 7 (Export Data3)
;
37$           WRITE,         Output:
                             CT:NEXT(28$);


;
;
;     Model statements for module:  BasicProcess.Decide 11 (Number Replication
4)
;
35$           BRANCH,        1:
                             If,RepNo==4,291$,Yes:
                             Else,292$,Yes;
291$          ASSIGN:        Number Replication 4.NumberOut True=Number
Replication 4.NumberOut True + 1:NEXT(38$);

292$          ASSIGN:        Number Replication 4.NumberOut False=Number
Replication 4.NumberOut False + 1:NEXT(36$);


;
;
;     Model statements for module:  AdvancedProcess.ReadWrite 8 (Export Data4)
;
38$           WRITE,         Output:
                             CT:NEXT(28$);


;
;
;     Model statements for module:  BasicProcess.Decide 12 (Number Replication
5)
;
36$           BRANCH,        1:
                             If,RepNo==5,293$,Yes:
                             Else,294$,Yes;
293$          ASSIGN:        Number Replication 5.NumberOut True=Number
Replication 5.NumberOut True + 1:NEXT(39$);

294$          ASSIGN:        Number Replication 5.NumberOut False=Number
Replication 5.NumberOut False + 1:NEXT(40$);


;
;
;     Model statements for module:  AdvancedProcess.ReadWrite 9 (Export Data5)
;
39$           WRITE,         Output:
                             CT:NEXT(28$);


;
;
```

```
;     Model statements for module:  BasicProcess.Decide 13 (Number Replication
6)
;
40$           BRANCH,        1:
                             If,RepNo==6,295$,Yes:
                             Else,296$,Yes;
295$          ASSIGN:        Number Replication 6.NumberOut True=Number
Replication 6.NumberOut True + 1:NEXT(41$);

296$          ASSIGN:        Number Replication 6.NumberOut False=Number
Replication 6.NumberOut False + 1:NEXT(42$);


;
;
;     Model statements for module:  AdvancedProcess.ReadWrite 10 (Export Data6)
;
41$           WRITE,         Output:
                             CT:NEXT(28$);


;
;
;     Model statements for module:  BasicProcess.Decide 14 (Number Replication
7)
;
42$           BRANCH,        1:
                             If,RepNo==7,297$,Yes:
                             Else,298$,Yes;
297$          ASSIGN:        Number Replication 7.NumberOut True=Number
Replication 7.NumberOut True + 1:NEXT(45$);

298$          ASSIGN:        Number Replication 7.NumberOut False=Number
Replication 7.NumberOut False + 1:NEXT(43$);


;
;
;     Model statements for module:  AdvancedProcess.ReadWrite 11 (Export Data7)
;
45$           WRITE,         Output:
                             CT:NEXT(28$);


;
;
;     Model statements for module:  BasicProcess.Decide 15 (Number Replication
8)
;
43$           BRANCH,        1:
                             If,RepNo==8,299$,Yes:
                             Else,300$,Yes;
299$          ASSIGN:        Number Replication 8.NumberOut True=Number
Replication 8.NumberOut True + 1:NEXT(46$);

300$          ASSIGN:        Number Replication 8.NumberOut False=Number
Replication 8.NumberOut False + 1:NEXT(44$);


;
```

```
;
;     Model statements for module:  AdvancedProcess.ReadWrite 12 (Export Data8)
;
46$           WRITE,          Output:
                              CT:NEXT(28$);


;
;
;     Model statements for module:  BasicProcess.Decide 16 (Number Replication
9)
;
44$           BRANCH,         1:
                              If,RepNo==9,301$,Yes:
                              Else,302$,Yes;
301$          ASSIGN:         Number Replication 9.NumberOut True=Number
Replication 9.NumberOut True + 1:NEXT(47$);

302$          ASSIGN:         Number Replication 9.NumberOut False=Number
Replication 9.NumberOut False + 1:NEXT(28$);


;
;
;     Model statements for module:  AdvancedProcess.ReadWrite 13 (Export Data9)
;
47$           WRITE,          Output:
                              CT:NEXT(28$);


;
;
;     Model statements for module:  BasicProcess.Dispose 4 (Dispose 4)
;
25$           ASSIGN:         Dispose 4.NumberOut=Dispose 4.NumberOut + 1;
303$          DISPOSE:        Yes;


;
;
;     Model statements for module:  BasicProcess.Create 4 (Simulation Start Up)
;

304$          CREATE,         1,MinutesToBaseTime(0.0),Entity
1:MinutesToBaseTime(EXPO(1)),1:NEXT(305$);

305$          ASSIGN:         Simulation Start Up.NumberOut=Simulation Start
Up.NumberOut + 1:NEXT(18$);


;
;
;     Model statements for module:  AdvancedProcess.ReadWrite 1 (Read TBA_X)
;
18$           READ,           MeanTBA_X:
                              ArrivalRate:NEXT(29$);


;
;
```

```
;     Model statements for module:  AdvancedProcess.ReadWrite 3 (Read Part Y)
;
29$           READ,          MeanTBA_Y:
                             ArrivalRate_Y:NEXT(30$);


;
;
;     Model statements for module:  AdvancedProcess.ReadWrite 5 (Read  Z)
;
30$           READ,          MeanTBA_Z:
                             ArrivalRate_Z:NEXT(19$);


;
;
;     Model statements for module:  BasicProcess.Assign 4 (Variables)
;
19$           ASSIGN:        RepNo=NREP:
                             LagY=LagValue:
                             LagZ=LagValue:
                             LagX=LagValue:
                             NoOfExists=0:
                             BatchNo=0:
                             BatchSize=0:
                             BatchCT=0:
                             DesiredTotalPartType=50000:
                             LagValue=25:
                             TruncationPoint=30000:NEXT(20$);


;
;
;     Model statements for module:  BasicProcess.Dispose 2 (Dispose 2)
;
20$           ASSIGN:        Dispose 2.NumberOut=Dispose 2.NumberOut + 1;
308$          DISPOSE:       Yes;
```

.exp file:

```
PROJECT,       "Unnamed
Project","tanushree.salvi@gmail.com",,,No,Yes,Yes,Yes,No,No,No,No,No,No;

ATTRIBUTES:    Type,DATATYPE(Real):
               ProcessTime,DATATYPE(Real):
               CT,DATATYPE(Real):
               ArrivalTime,DATATYPE(Real);

FILES:         Output,"T:\Thesis\Job Shop\Output.txt",Sequential,Free
Format,Dispose,,Hold:
               MeanTBA_X,"T:\Thesis\Job Shop\MeanTBA_X.txt",Sequential,Free
Format,Dispose,,Hold:
               MeanTBA_Y,"T:\Thesis\Job Shop\MeanTBA_Y.txt",Sequential,Free
Format,Dispose,,Hold:
               MeanTBA_Z,"T:\Thesis\Job Shop\MeanTBA_Z.txt",Sequential,Free
Format,Dispose,,Hold;

VARIABLES:     Tool Group 4.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
               Dispose 2.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
               Number Replication 1.NumberOut
False,CLEAR(Statistics),CATEGORY("Exclude"):
               Dispose 5.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
               LagX,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
               LagY,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
               LagZ,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
               Number Replication 3.NumberOut
True,CLEAR(Statistics),CATEGORY("Exclude"):
               Number Replication 6.NumberOut
False,CLEAR(Statistics),CATEGORY("Exclude"):
               DesiredTotalPartType,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
               BatchNo,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
               Number Replication 7.NumberOut
True,CLEAR(Statistics),CATEGORY("Exclude"):
               Tool Group 2.WIP,CLEAR(System),CATEGORY("Exclude-
Exclude"),DATATYPE(Real):
               Tool Group 3.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
               Number Replication 2.NumberOut
False,CLEAR(Statistics),CATEGORY("Exclude"):
               Dispose 4.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
               Dispose rest.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
               Number Replication 7.NumberOut
False,CLEAR(Statistics),CATEGORY("Exclude"):
               Part Y Arrival.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
               BatchSize,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
               Number Replication 4.NumberOut
True,CLEAR(Statistics),CATEGORY("Exclude"):
               ArrivalRate_Y,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
               ArrivalRate_Z,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
               Tool Group 4.WIP,CLEAR(System),CATEGORY("Exclude-
Exclude"),DATATYPE(Real):
```

```
              At Truncation Point.NumberOut
True,CLEAR(Statistics),CATEGORY("Exclude"):
              Tool Group 1.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
              Number Replication 3.NumberOut
False,CLEAR(Statistics),CATEGORY("Exclude"):
              Tool Group 3.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
              LagValue,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
              Number Replication 8.NumberOut
True,CLEAR(Statistics),CATEGORY("Exclude"):
              BatchCT,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
              ArrivalRate,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
              Number Replication 1.NumberOut
True,CLEAR(Statistics),CATEGORY("Exclude"):
              NoOfExists,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
              Number Replication 8.NumberOut
False,CLEAR(Statistics),CATEGORY("Exclude"):
              Part Z Arrival.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              Tool Group 1.WIP,CLEAR(System),CATEGORY("Exclude-
Exclude"),DATATYPE(Real):
              Tool Group 2.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              Number Replication 5.NumberOut
True,CLEAR(Statistics),CATEGORY("Exclude"):
              Number Replication 4.NumberOut
False,CLEAR(Statistics),CATEGORY("Exclude"):
              Simulation Start
Up.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              Lag Long enough?.NumberOut
True,CLEAR(Statistics),CATEGORY("Exclude"):
              Tool Group 3.WIP,CLEAR(System),CATEGORY("Exclude-
Exclude"),DATATYPE(Real):
              At Truncation Point.NumberOut
False,CLEAR(Statistics),CATEGORY("Exclude"):
              Number Replication 2.NumberOut
True,CLEAR(Statistics),CATEGORY("Exclude"):
              Number Replication 9.NumberOut
True,CLEAR(Statistics),CATEGORY("Exclude"):
              Number Replication 5.NumberOut
False,CLEAR(Statistics),CATEGORY("Exclude"):
              Lag Long enough?.NumberOut
False,CLEAR(Statistics),CATEGORY("Exclude"):
              TruncationPoint,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
              Number Replication 9.NumberOut
False,CLEAR(Statistics),CATEGORY("Exclude"):
              RepNo,CLEAR(System),CATEGORY("User Specified-User
Specified"),DATATYPE(Real):
              Tool Group 2.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
              Number Replication 6.NumberOut
True,CLEAR(Statistics),CATEGORY("Exclude"):
              Tool Group 4.NumberIn,CLEAR(Statistics),CATEGORY("Exclude"):
              Tool Group 1.NumberOut,CLEAR(Statistics),CATEGORY("Exclude"):
              Part X Arrival.NumberOut,CLEAR(Statistics),CATEGORY("Exclude");

QUEUES:       Tool Group 1.Queue,FIFO,,AUTOSTATS(Yes,,):
              Tool Group 2.Queue,FIFO,,AUTOSTATS(Yes,,):
              Tool Group 3.Queue,FIFO,,AUTOSTATS(Yes,,):
```

```
               Tool Group 4.Queue,FIFO,,AUTOSTATS(Yes,,);

PICTURES:      Picture.Airplane:
               Picture.Green Ball:
               Picture.Blue Page:
               Picture.Telephone:
               Picture.Blue Ball:
               Picture.Yellow Page:
               Picture.EMail:
               Picture.Yellow Ball:
               Picture.Bike:
               Picture.Report:
               Picture.Van:
               Picture.Widgets:
               Picture.Envelope:
               Picture.Fax:
               Picture.Truck:
               Picture.Person:
               Picture.Letter:
               Picture.Box:
               Picture.Woman:
               Picture.Package:
               Picture.Man:
               Picture.Diskette:
               Picture.Boat:
               Picture.Red Page:
               Picture.Ball:
               Picture.Green Page:
               Picture.Red Ball;

RESOURCES:     Operator
1,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes,,):
               Operator
2,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes,,):
               Operator
3,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes,,):
               Operator
4,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes,,):
               Machine
1,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes,,):
               Machine
2,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes,,):
               Machine
3,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes,,):
               Machine
4,Capacity(1),,,COST(0.0,0.0,0.0),CATEGORY(Resources),,AUTOSTATS(Yes,,);

STATIONS:      Collect Stats,,,Collect Stats,AUTOSTATS(Yes,,):
               TG 1,,,TG 1,AUTOSTATS(Yes,,):
               TG 2,,,TG 2,AUTOSTATS(Yes,,):
               TG 3,,,TG 3,AUTOSTATS(Yes,,):
               TG 4,,,TG 4,AUTOSTATS(Yes,,);

SEQUENCES:     Sequence for Part X,TG 1,,,,ProcessTime=NORM(13,0.5)&TG
2,,,,ProcessTime=NORM(10.4,0.4)&TG 3,,,,ProcessTime=
               NORM(5.2,0.2)&Collect Stats:
               Sequence for Part Y,TG 2,,,,ProcessTime=NORM(10.4,0.4)&TG
1,,,,ProcessTime=NORM(3.9,0.15)&TG 4,,,,ProcessTime=
               NORM(6.5,0.25)&TG 3,,,,ProcessTime=NORM(7.8,0.3)&Collect Stats:
```

```
               Sequence for Part Z,TG 1,,,,ProcessTime=NORM(5.2,0.2)&TG
2,,,,ProcessTime=NORM(9.1,0.35)&TG 4,,,,ProcessTime=
               NORM(3.9,0.5)&Collect Stats;

COUNTERS:      Exists,,,,DATABASE(,"Count","User Specified","Exists");

REPLICATE,
1,,,Yes,Yes,,NC(Exists)>DesiredTotalPartType,,24,Minutes,No,No,,,Yes,No;

ENTITIES:      Entity 1,Picture.Report,0.0,0.0,0.0,0.0,0.0,0.0,AUTOSTATS(Yes,,);

ACTIVITYAREAS: Collect Stats,0,,AUTOSTATS(Yes,,):
               TG 1,0,,AUTOSTATS(Yes,,):
               TG 2,0,,AUTOSTATS(Yes,,):
               TG 3,0,,AUTOSTATS(Yes,,):
               TG 4,0,,AUTOSTATS(Yes,,);
```

APPENDIX C

MATLAB PROGRAMMING CODE FOR MINI-FAB MODEL

C. MATLAB programming code for mini-fab model:

```
function [] = Code_M ()
data= dlmread ('XData.txt');
hist(data);
%REPLICATION 1----------------------------------------------------
%Reading output from Arena in Excel Sheet
output = dlmread('XData.txt','\t','A1..A5001');
xlswrite ('T:\Thesis\Model\Experiments\Individual Cycle
Time_M',output,1,'B2');
%Calculating the moments
Mean = mean(output);
Stddev = std(output);
Skewness = skewness(output);
Kurtosis = kurtosis(output);
C = sort(output);%sorting the numbers in ascending order
count = numel(output);%counting number of elements in the output
percentile = prctile(output, [5:5:95]);
Cell= 'B7';
xlswrite('T:\Thesis\Model\Experiments\Matlab_Output_M1.xlsx',percentile
',1,Cell);
% hist(output);
A= {Mean; Stddev; Skewness; Kurtosis};
x1Range = 'B2';
xlswrite
('T:\Thesis\Model\Experiments\Matlab_Output_M1.xlsx',A,1,x1Range);
%REPLICATION 2----------------------------------------------------
%Reading output from Arena in Excel Sheet
output = dlmread('XData.txt','\t','A50002..A10002');
xlswrite ('T:\Thesis\Model\Experiments\Individual Cycle
Time_M',output,1,'C2');
%Calculating the moments
Mean = mean(output);
Stddev = std(output);
Skewness = skewness(output);
Kurtosis = kurtosis(output);
C = sort(output);%sorting the numbers in ascending order
count = numel(output); %counting number of elements in the output
 percentile = prctile(output, [5:5:95]);
 Cell= 'C7';
xlswrite('T:\Thesis\Model\Experiments\Matlab_Output_M.xlsx',percentile'
,1,Cell);
% hist(output);
A= {Mean; Stddev; Skewness; Kurtosis};
x1Range = 'C2';
xlswrite
('T:\Thesis\Model\Experiments\Matlab_Output_M.xlsx',A,1,x1Range);
%REPLICATION 3--------------------------------------------------------
%Reading output from Arena in Excel Sheet
output = dlmread('XData.txt','\t','A10003..A15003');
xlswrite ('T:\Thesis\Model\Experiments\Individual Cycle
Time_M',output,1,'D2');
%Calculating the moments
Mean = mean(output);
```

```
Stddev = std(output);
Skewness = skewness(output);
Kurtosis = kurtosis(output);
C = sort(output);%sorting the numbers in ascending order
count = numel(output); %counting number of elements in the output
 percentile = prctile(output, [5:5:95]);
 Cell= 'D7';
xlswrite('T:\Thesis\Model\Experiments\Matlab_Output_M.xlsx',percentile'
,1,Cell);
% hist(output);
A= {Mean; Stddev; Skewness; Kurtosis};
x1Range = 'D2';
xlswrite
('T:\Thesis\Model\Experiments\Matlab_Output_M.xlsx',A,1,x1Range);
%REPLICATION 4-----------------------------------------------------------
-
%Reading output from Arena in Excel Sheet
output = dlmread('XData.txt','\t','A15004..A20004');
xlswrite ('T:\Thesis\Model\Experiments\Individual Cycle
Time_M',output,1,'E2');
%Calculating the moments
Mean = mean(output);
Stddev = std(output);
Skewness = skewness(output);
Kurtosis = kurtosis(output);
C = sort(output);%sorting the numbers in ascending order
count = numel(output); %counting number of elements in the output
 percentile = prctile(output, [5:5:95]);
 Cell= 'E7';
xlswrite('T:\Thesis\Model\Experiments\Matlab_Output_M.xlsx',percentile'
,1,Cell);
% hist(output);
A= {Mean; Stddev; Skewness; Kurtosis};
x1Range = 'E2';
xlswrite
('T:\Thesis\Model\Experiments\Matlab_Output_M.xlsx',A,1,x1Range);
%REPLICATION 5-----------------------------------------------------------
-
%Reading output from Arena in Excel Sheet
output = dlmread('XData.txt','\t','A20005..A25005');
xlswrite ('T:\Thesis\Model\Experiments\Individual Cycle
Time_M',output,1,'F2');
%Calculating the moments
Mean = mean(output);
Stddev = std(output);
Skewness = skewness(output);
Kurtosis = kurtosis(output);
C = sort(output);%sorting the numbers in ascending order
count = numel(output); %counting number of elements in the output
 percentile = prctile(output, [5:5:95]);
 Cell= 'F7';
xlswrite('T:\Thesis\Model\Experiments\Matlab_Output_M.xlsx',percentile'
,1,Cell);
% hist(output);
```

```
A= {Mean; Stddev; Skewness; Kurtosis};
x1Range = 'F2';
xlswrite
('T:\Thesis\Model\Experiments\Matlab_Output_M.xlsx',A,1,x1Range);
%REPLICATION 6--------------------------------------------------------
-
%Reading output from Arena in Excel Sheet
output = dlmread('XData.txt','\t','A25006..A30006');
xlswrite ('T:\Thesis\Model\Experiments\Individual Cycle
Time_M',output,1,'G2');
%Calculating the moments
Mean = mean(output);
Stddev = std(output);
Skewness = skewness(output);
Kurtosis = kurtosis(output);
C = sort(output);%sorting the numbers in ascending order
count = numel(output); %counting number of elements in the output
 percentile = prctile(output, [5:5:95]);
 Cell= 'G7';
xlswrite('T:\Thesis\Model\Experiments\Matlab_Output_M.xlsx',percentile'
,1,Cell);
% hist(output);
A= {Mean; Stddev; Skewness; Kurtosis};
x1Range = 'G2';
xlswrite
('T:\Thesis\Model\Experiments\Matlab_Output_M.xlsx',A,1,x1Range);
%REPLICATION 7--------------------------------------------------------
-
%Reading output from Arena in Excel Sheet
output = dlmread('XData.txt','\t','A30007..A35007');
xlswrite ('T:\Thesis\Model\Experiments\Individual Cycle
Time_M',output,1,'H2');
%Calculating the moments
Mean = mean(output);
Stddev = std(output);
Skewness = skewness(output);
Kurtosis = kurtosis(output);
C = sort(output);%sorting the numbers in ascending order
count = numel(output); %counting number of elements in the output
 percentile = prctile(output, [5:5:95]);
 Cell= 'H7';
xlswrite('T:\Thesis\Model\Experiments\Matlab_Output_M.xlsx',percentile'
,1,Cell);
% hist(output);
A= {Mean; Stddev; Skewness; Kurtosis};
x1Range = 'H2';
xlswrite
('T:\Thesis\Model\Experiments\Matlab_Output_M.xlsx',A,1,x1Range);
%REPLICATION 8--------------------------------------------------------
-
%Reading output from Arena in Excel Sheet
output = dlmread('XData.txt','\t','A35008..A40008');
xlswrite ('T:\Thesis\Model\Experiments\Individual Cycle
Time_M',output,1,'I2');
```

```
%Calculating the moments
Mean = mean(output);
Stddev = std(output);
Skewness = skewness(output);
Kurtosis = kurtosis(output);
C = sort(output);%sorting the numbers in ascending order
count = numel(output); %counting number of elements in the output
 percentile = prctile(output, [5:5:95]);
 Cell= 'I7';
xlswrite('T:\Thesis\Model\Experiments\Matlab_Output_M.xlsx',percentile'
,1,Cell);
% hist(output);
A= {Mean; Stddev; Skewness; Kurtosis};
x1Range = 'I2';
xlswrite
('T:\Thesis\Model\Experiments\Matlab_Output_M.xlsx',A,1,x1Range);
%REPLICATION 9-------------------------------------------------------
-
%Reading output from Arena in Excel Sheet
output = dlmread('XData.txt','\t','A40009..A45009');
xlswrite ('T:\Thesis\Model\Experiments\Individual Cycle
Time_M',output,1,'J2');
%Calculating the moments
Mean = mean(output);
Stddev = std(output);
Skewness = skewness(output);
Kurtosis = kurtosis(output);
C = sort(output);%sorting the numbers in ascending order
count = numel(output); %counting number of elements in the output
 percentile = prctile(output, [5:5:95]);
 Cell= 'J7';
xlswrite('T:\Thesis\Model\Experiments\Matlab_Output_M.xlsx',percentile'
,1,Cell);
% hist(output);
A= {Mean; Stddev; Skewness; Kurtosis};
x1Range = 'J2';
xlswrite
('T:\Thesis\Model\Experiments\Matlab_Output_M.xlsx',A,1,x1Range);
%REPLICATION 10------------------------------------------------------
--
%Reading output from Arena in Excel Sheet
output = dlmread('XData.txt','\t','A45010..A50010');
xlswrite ('T:\Thesis\Model\Experiments\Individual Cycle
Time_M',output,1,'K2');
%Calculating the moments
Mean = mean(output);
Stddev = std(output);
Skewness = skewness(output);
Kurtosis = kurtosis(output);
C = sort(output);%sorting the numbers in ascending order
count = numel(output); %counting number of elements in the output
 percentile = prctile(output, [5:5:95]);
 Cell= 'K7';
```

```
xlswrite('T:\Thesis\Model\Experiments\Matlab_Output_M.xlsx',percentile'
,1,Cell);
% hist(output);
A= {Mean; Stddev; Skewness; Kurtosis};
x1Range = 'K2';
xlswrite
('T:\Thesis\Model\Experiments\Matlab_Output_M.xlsx',A,1,x1Range);
%REPLICATION 11--------------------------------------------------------
--
%Reading output from Arena in Excel Sheet
output = dlmread('XData.txt','\t','A50011..A55011');
xlswrite ('T:\Thesis\Model\Experiments\Individual Cycle
Time_M',output,1,'L2');
%Calculating the moments
Mean = mean(output);
Stddev = std(output);
Skewness = skewness(output);
Kurtosis = kurtosis(output);
C = sort(output);%sorting the numbers in ascending order
count = numel(output); %counting number of elements in the output
 percentile = prctile(output, [5:5:95]);
 Cell= 'L7';
xlswrite('T:\Thesis\Model\Experiments\Matlab_Output_M.xlsx',percentile'
,1,Cell);
% hist(output);
A= {Mean; Stddev; Skewness; Kurtosis};
x1Range = 'L2';
xlswrite
('T:\Thesis\Model\Experiments\Matlab_Output_M.xlsx',A,1,x1Range);
%REPLICATION 12--------------------------------------------------------
--
%Reading output from Arena in Excel Sheet
output = dlmread('XData.txt','\t','A55012..A60012');
xlswrite ('T:\Thesis\Model\Experiments\Individual Cycle
Time_M',output,1,'M2');
%Calculating the moments
Mean = mean(output);
Stddev = std(output);
Skewness = skewness(output);
Kurtosis = kurtosis(output);
C = sort(output);%sorting the numbers in ascending order
count = numel(output); %counting number of elements in the output
 percentile = prctile(output, [5:5:95]);
 Cell= 'M7';
xlswrite('T:\Thesis\Model\Experiments\Matlab_Output_M.xlsx',percentile'
,1,Cell);
% hist(output);
A= {Mean; Stddev; Skewness; Kurtosis};
x1Range = 'M2';
xlswrite
('T:\Thesis\Model\Experiments\Matlab_Output_M.xlsx',A,1,x1Range);
end
```

APPENDIX D

MATLAB PROGRAMMING CODE FOR JOB SHOP MODEL

D. MATLAB programming code for job shop model:

```
function [] = Code_JS ()
data= dlmread ('Output.txt');
hist(data);
%REPLICATION 1-----------------------------------------------------
%Reading output from Arena in Excel Sheet
output = dlmread('Output.txt','\t','A1..A50001');
xlswrite ('Individual Cycle Time',output,1,'B2');
%Calculating the moments
Mean = mean(output);
Stddev = std(output);
Skewness = skewness(output);
Kurtosis = kurtosis(output);
C = sort(output);%sorting the numbers in ascending order
count = numel(output);%counting number of elements in the output
percentile = prctile(output, [5:5:95]);
Cell= 'B7';
xlswrite('T:\Thesis\Job
Shop\Experiments\Matlab_Output_JS.xlsx',percentile',1,Cell);
% hist(output);
A= {Mean; Stddev; Skewness; Kurtosis};
x1Range = 'B2';
xlswrite ('T:\Thesis\Job
Shop\Experiments\Matlab_Output_JS.xlsx',A,1,x1Range);
%REPLICATION 2-----------------------------------------------------
%Reading output from Arena in Excel Sheet
output = dlmread('Output.txt','\t','A50002..A100002');
xlswrite ('Individual Cycle Time',output,1,'C2');
%Calculating the moments
Mean = mean(output);
Stddev = std(output);
Skewness = skewness(output);
Kurtosis = kurtosis(output);
C = sort(output);%sorting the numbers in ascending order
count = numel(output); %counting number of elements in the output
 percentile = prctile(output, [5:5:95]);
 Cell= 'C7';
xlswrite('T:\Thesis\Job
Shop\Experiments\Matlab_Output_JS.xlsx',percentile',1,Cell);
% hist(output);
A= {Mean; Stddev; Skewness; Kurtosis};
x1Range = 'C2';
xlswrite ('T:\Thesis\Job
Shop\Experiments\Matlab_Output_JS.xlsx',A,1,x1Range);
%REPLICATION 3---------------------------------------------------------
%Reading output from Arena in Excel Sheet
output = dlmread('Output.txt','\t','A100003..A150003');
xlswrite ('Individual Cycle Time',output,1,'D2');
%Calculating the moments
Mean = mean(output);
Stddev = std(output);
Skewness = skewness(output);
Kurtosis = kurtosis(output);
```

```
C = sort(output);%sorting the numbers in ascending order
count = numel(output); %counting number of elements in the output
 percentile = prctile(output, [5:5:95]);
 Cell= 'D7';
xlswrite('T:\Thesis\Job
Shop\Experiments\Matlab_Output_JS.xlsx',percentile',1,Cell);
% hist(output);
A= {Mean; Stddev; Skewness; Kurtosis};
x1Range = 'D2';
xlswrite ('T:\Thesis\Job
Shop\Experiments\Matlab_Output_JS.xlsx',A,1,x1Range);
%REPLICATION 4---------------------------------------------------------
-
%Reading output from Arena in Excel Sheet
output = dlmread('Output.txt','\t','A150004..A200004');
xlswrite ('Individual Cycle Time',output,1,'E2');
%Calculating the moments
Mean = mean(output);
Stddev = std(output);
Skewness = skewness(output);
Kurtosis = kurtosis(output);
C = sort(output);%sorting the numbers in ascending order
count = numel(output); %counting number of elements in the output
 percentile = prctile(output, [5:5:95]);
 Cell= 'E7';
xlswrite('T:\Thesis\Job
Shop\Experiments\Matlab_Output_JS.xlsx',percentile',1,Cell);
% hist(output);
A= {Mean; Stddev; Skewness; Kurtosis};
x1Range = 'E2';
xlswrite ('T:\Thesis\Job
Shop\Experiments\Matlab_Output_JS.xlsx',A,1,x1Range);
%REPLICATION 5---------------------------------------------------------
-
%Reading output from Arena in Excel Sheet
output = dlmread('Output.txt','\t','A200005..A250005');
xlswrite ('Individual Cycle Time',output,1,'F2');
%Calculating the moments
Mean = mean(output);
Stddev = std(output);
Skewness = skewness(output);
Kurtosis = kurtosis(output);
C = sort(output);%sorting the numbers in ascending order
count = numel(output); %counting number of elements in the output
 percentile = prctile(output, [5:5:95]);
 Cell= 'F7';
xlswrite('T:\Thesis\Job
Shop\Experiments\Matlab_Output_JS.xlsx',percentile',1,Cell);
% hist(output);
A= {Mean; Stddev; Skewness; Kurtosis};
x1Range = 'F2';
xlswrite ('T:\Thesis\Job
Shop\Experiments\Matlab_Output_JS.xlsx',A,1,x1Range);
```

```
%REPLICATION 6----------------------------------------------------------
-
%Reading output from Arena in Excel Sheet
output = dlmread('Output.txt','\t','A250006..A300006');
xlswrite ('Individual Cycle Time',output,1,'G2');
%Calculating the moments
Mean = mean(output);
Stddev = std(output);
Skewness = skewness(output);
Kurtosis = kurtosis(output);
C = sort(output);%sorting the numbers in ascending order
count = numel(output); %counting number of elements in the output
 percentile = prctile(output, [5:5:95]);
 Cell= 'G7';
xlswrite('T:\Thesis\Job
Shop\Experiments\Matlab_Output_JS.xlsx',percentile',1,Cell);
% hist(output);
A= {Mean; Stddev; Skewness; Kurtosis};
x1Range = 'G2';
xlswrite ('T:\Thesis\Job
Shop\Experiments\Matlab_Output_JS.xlsx',A,1,x1Range);
%REPLICATION 7----------------------------------------------------------
-
%Reading output from Arena in Excel Sheet
output = dlmread('Output.txt','\t','A300007..A350007');
xlswrite ('Individual Cycle Time',output,1,'H2');
%Calculating the moments
Mean = mean(output);
Stddev = std(output);
Skewness = skewness(output);
Kurtosis = kurtosis(output);
C = sort(output);%sorting the numbers in ascending order
count = numel(output); %counting number of elements in the output
 percentile = prctile(output, [5:5:95]);
 Cell= 'H7';
xlswrite('T:\Thesis\Job
Shop\Experiments\Matlab_Output_JS.xlsx',percentile',1,Cell);
% hist(output);
A= {Mean; Stddev; Skewness; Kurtosis};
x1Range = 'H2';
xlswrite ('T:\Thesis\Job
Shop\Experiments\Matlab_Output_JS.xlsx',A,1,x1Range);
%REPLICATION 8----------------------------------------------------------
-
%Reading output from Arena in Excel Sheet
output = dlmread('Output.txt','\t','A350008..A400008');
xlswrite ('Individual Cycle Time',output,1,'I2');
%Calculating the moments
Mean = mean(output);
Stddev = std(output);
Skewness = skewness(output);
Kurtosis = kurtosis(output);
C = sort(output);%sorting the numbers in ascending order
count = numel(output); %counting number of elements in the output
```

```
 percentile = prctile(output, [5:5:95]);
 Cell= 'I7';
xlswrite('T:\Thesis\Job
Shop\Experiments\Matlab_Output_JS.xlsx',percentile',1,Cell);
% hist(output);
A= {Mean; Stddev; Skewness; Kurtosis};
x1Range = 'I2';
xlswrite ('T:\Thesis\Job
Shop\Experiments\Matlab_Output_JS.xlsx',A,1,x1Range);
%REPLICATION 9-------------------------------------------------------
-
%Reading output from Arena in Excel Sheet
output = dlmread('Output.txt','\t','A400009..A450009');
xlswrite ('Individual Cycle Time',output,1,'J2');
%Calculating the moments
Mean = mean(output);
Stddev = std(output);
Skewness = skewness(output);
Kurtosis = kurtosis(output);
C = sort(output);%sorting the numbers in ascending order
count = numel(output); %counting number of elements in the output
 percentile = prctile(output, [5:5:95]);
 Cell= 'J7';
xlswrite('T:\Thesis\Job
Shop\Experiments\Matlab_Output_JS.xlsx',percentile',1,Cell);
% hist(output);
A= {Mean; Stddev; Skewness; Kurtosis};
x1Range = 'J2';
xlswrite ('T:\Thesis\Job
Shop\Experiments\Matlab_Output_JS.xlsx',A,1,x1Range);
end
```