AN INTERVAL BASED APPROACH TO MODEL INPUT UNCERTAINTY IN DISCRETE-EVENT SIMULATION

by

OLA GHAZI Y. BATARSEH

B.Sc. University of Jordan, 2006
M.S. University of Central Florida, 2008

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Industrial Engineering and Management Systems
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2010

Major Professor: Yan Wang

ABSTRACT

The objective of this research is to increase the robustness of discrete-event simulation (DES) when input uncertainties associated models and parameters are present. Input uncertainties in simulation have different sources, including lack of data, conflicting information and beliefs, lack of introspection, measurement errors, and lack of information about dependency. A reliable solution is obtained from a simulation mechanism that accounts for these uncertainty components in simulation.

An interval-based simulation (IBS) mechanism based on imprecise probabilities is proposed, where the statistical distribution parameters in simulation are intervals instead of precise real numbers. This approach incorporates variability and uncertainty in systems. In this research, a standard procedure to estimate interval parameters of probability distributions is developed based on the measurement of simulation robustness. New mechanisms based on the inverse transform to generate interval random variates are proposed. A generic approach to specify the required replication length to achieve a desired level of robustness is derived. Furthermore, three simulation clock advancement approaches in the interval-based simulation are investigated. A library of Java-based IBS toolkits that simulates queueing systems is developed to demonstrate the new proposed reliable simulation. New interval statistics for interval data analysis are proposed to support decision making. To assess the performance of the IBS, we developed an interval-based metamodel for automated material handling systems, which generates interval performance measures that are more reliable and computationally more efficient than traditional DES simulation results.

Dedication

To my beloved parents Ghazi and Maha

and my dearest sister Majd

"Blessed is the man who trusts in the Lord and has made the Lord his hope and

confidence."

ACKNOWLEDGMENTS

Above all others, I would like to thank my exceptional family to whom this dissertation is dedicated to. Their endless support, faith, and love have aided and encouraged me throughout this endeavor. My ultimate thanks and sincerest appreciation are devoted to my parents, my father Ghazi and my mother Maha, and my siblings, Majd and Mina, Essa, Noor, Mahd, Mina, and Ameer. I owe a special thank to my eldest sister and dearest friend, Majd, for being there for me, providing inspiration and most importantly having fun times together during this journey. I would not be the person that I am today without my family.

Academically, I am very grateful to my advisor, Dr. Yan Wang. I have been amazingly fortunate to have Dr. Wang as my advisor. He generously provided me with long hours of in–depth discussions, thorough revisions and editing and wise supervision. His enlightening discussions, encouraging words, and patience made this work possible.

Likewise, I would like to express my appreciation to my committee members. I am sincerely grateful and highly indebted to Dr. Dima Nazzal for her thorough help, constant support, encouragement, and friendship. I thank Dr. Scott Ferson for offering academic advice in my research and for his willingness to debate research issues. I would like also to thank Dr. Mansooreh Mollaghsemi for her time and support.

Last but not least, I would like to thank my family and friends in Jordan and Orlando, and all those who trusted in me.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AMHS	Automated Material Handling Simulation
BMA	Bayesian Moving Average
cdf	Cumulative Distribution Function
DES	Discrete Event Simulation
EDF	Empirical Distribution Function
$Exp(\lambda)$	Exponential Distribution with a parameter mean of λ
$Exp([\underline{\lambda}, \overline{\lambda}])$	Exponential Distribution with an imprecise parameter mean of $[\underline{\lambda}, \overline{\lambda}]$
FIFO	First-in-First-out
IBS	Interval-based Simulation
IID	Independent and Identically Distributed
MLE	Maximum Likelihood Estimation
M/M/1	A notation to represent a single server that has unlimited queue capacity and infinite calling population, both arrivals and service are Poisson processes
[M]/[M]/1	A notation to represent a single server that has unlimited queue capacity and infinite calling population, both arrivals and service are Poisson processes with interval parameters
pdf	Probability Distribution Function
SOMC	Second-order Monte Carlo
$[\underline{x}, \overline{x}]$	Generalized Continuous Interval
$\{\underline{x}, \overline{x}\}$	Generalized Discrete Interval

CHAPTER 1:
INTRODUCTION

With the advancement of computational capabilities, simulation has become one of the most widely used operations research and management science techniques. Application areas for simulation are numerous and diverse, and it has been widely used in several applications, namely manufacturing systems, service organizations, military weapons systems, systems' logistics requirements, operating transportation systems, and supply chain management. Simulation is used to imitate systems, real-world facilities or processes, by the use of computers. DES can be viewed as a 10-step study, as listed in Figure 1-1 (Law, 2007).
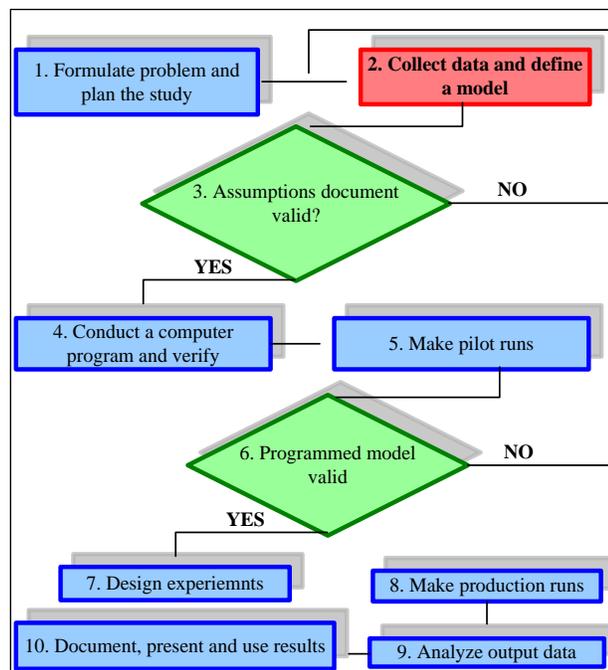
Figure 1-1: Steps is a DES Study

Figure 1-1 represents the steps that formulate a common simulation study. Attention must be paid to step 2 so that a reliable input analysis is guaranteed. In this step, the collection of data and modeling the input random variables using statistical distributions are performed. Traditionally, the maximum likelihood estimation (MLE) is used to estimate input parameters of the statistical distribution. This approach models statistical distributions with real-valued parameters. One of the main problems in this approach is the assumption that the input parameters are known with certitude. However, when uncertainty is significant in making decisions, it must be quantified and incorporated into simulations. In addition, the uncertainty must be interpreted through simulation predictions regardless of its source.

The existence of uncertainties in real-world systems would never make the input parameters of distribution functions known in certainty. The fact that MLEs are asymptotically normally distributed suggests a way of checking how sensitive a simulation output with respect to (w.r.t.) a particular input parameter. This characteristic allows us to build a confidence interval for the estimated parameter of the MLE. Consequently, simulation experiments are run with input parameters using the end-points of the confidence interval. If performance measures of interest are sensitive to the selected value of the input parameter, this implies that it is erroneous to make our decisions based upon one single-value input parameters.

The objective of this research is to create a new *reliable* interval-based discrete-event simulation mechanism. A simulation mechanism is reliable if the completeness and soundness of its results w.r.t. uncertainties can be verified. A *complete* solution includes all possible occurrences. A *sound* solution does not include impossible occurrences. The

proposed simulation mechanism allows us to verify the completeness and soundness of results easily, and at the same time computationally efficiently.

1.1   Statement of Problem

In traditional simulation mechanisms, uncertainties are modeled using probability distributions with real-valued parameters, (e.g., *Exponential*(1.0)). This uncertainty representation does not capture the total uncertainty components explicitly. The total uncertainty is composed of two components:

Variability: is due to the inherent randomness in the system. In literature, variability is also referred to as stochastic uncertainty, simulation uncertainty, aleatory uncertainty, and irreducible uncertainty. This component is irreducible even by additional measurements. The typical representation of variability is based on probability distributions.

Uncertainty: is due to the lack of perfect knowledge or enough information about the system. Uncertainty is also known as epistemic uncertainty, reducible uncertainty, and model form uncertainty. Since the uncertainty is caused by the lack of information about the system, it can be reduced by increasing our knowledge to fill the information gap.

The distinction between these types of uncertainty is important because, as mentioned above, uncertainty is reducible. Parameter uncertainty can be characterized as epistemic, because as the random variable sample size increases, the uncertainty about the parameter value is reduced. From a psychological point of view, epistemic uncertainty reflects the possibility of errors in our general knowledge. This has attracted

researchers to study how our neural systems respond to degrees of uncertainty in human decision-making (Hsu et al., 2005). Experiments have shown that there is a strong relationship between the level of uncertainties in decision-making and the activation level of our neural system. Thus, we can conclude that capturing the total uncertainty components in simulation supports *robust* decision making.

The classical simulation mechanism currently adopted in the modeling and simulation (M&S) industry lacks the ability to study systems with the consideration of the above two components separately where the input probability distributions and their parameters in these simulations are deterministic and precise. One can argue that this approach may lead to erroneous conclusions because only the variability component of the total uncertainty is represented, while the uncertainty component is ignored. The reliability of inputs, and thus, the outputs of these simulations is unknown. When simulations are used in system analysis with significant risks involved and where decisions made based on simulation results are sensitive to the uncertainty component, we need to incorporate the total uncertainty in simulations. The reliability of output measures and the robustness of decisions are expected to be improved when the total uncertainty is captured in inputs. Thus, how accurately a simulation model can predict is directly related to the reliability of input analysis.

There are many arguments (e.g. Walley, 1991; Draper, 1995; & Zimmermann, 2000) which deliberate the sources of uncertainties and support the admission of the total uncertainty. The most important of those related to simulation include:

- *Lack of data:* the parameters of probability distributions and distribution types are uncertain when the sample size of data for input analysis is small. In some situations,

it may not be possible to collect enough data on the random variables of interest. The lack of enough information introduces errors in simulation and requires the analyst to find new ways to describe the associated uncertainty more rigorously.

- *Conflicting information*: if there are multiple sources of information, the analyst may face conflicts within the data sources. It is not appropriate to draw a simple conclusion of distributions from several pieces of contradictory evidence. The absence of relevant information has to be properly modeled and incorporated in the simulation.

- *Conflicting beliefs*: when data is not available, the analyst usually depends on expert judgments and opinions. The information obtained from experts is subjective due to the diversity of their past experiences, which can lead to inconsistent observations.

- *Lack of introspection*: in some cases, the analyst cannot afford the necessary time to think deliberately about an uncertain event or an accurate description of physical systems. The lack of introspection increases the risk of inaccurate simulation of the systems under study. The associated uncertainties have to be reflected in the simulation input distributions to receive more reliable outputs.

- *Measurement errors*: all measurements are approximated values (instead of true values) due to the limitations of measuring devices and environment, process of measurement, and human errors. The uncertainties associated with the collected quantities should be addressed in input analysis in order to give more reliable outputs.

- *Lack of information about dependency*: lack of evidence about the dependency among factors and variables as well as unknown time dependency of these factors contribute to increase simulation models' uncertainties. For example, when a queueing system

has dependencies between arrival and service patterns, simply modeling it as an M/M/1 queue with arrival and service rates as the means of the available data would result in ignoring the dependencies. The estimation about the waiting time will be inaccurate. The consideration of unknown dependency among variables will build more reliable simulation models.

The sources of uncertainties mentioned above draw our attention to the importance of including the uncertainty component in simulation. It is ultimately crucial to understand the impact of capturing the total uncertainty components on robust decision making. The two components of the total uncertainty have to be represented explicitly in simulation. In literature, uncertainties associated with distribution types are referred to as model uncertainties, and those with parameter values as parameter uncertainties. Therefore, in this document, *input uncertainties* refer to both. This research promises to provide a new reliable DES mechanism that incorporates input uncertainties in simulation.

1.2   Motivation of this Research

Simulation model performance measures are a function of a particular input parameter value. Traditionally, simulation analysts use heuristics or graphical procedures to fit a statistical distribution for the available data. However, the analysts do not know with absolute certainty if they are using the right model. Model uncertainty is prevalent when using these methods (e.g., *Exponential*(1.0) vs. *Uniform*([0.5,1.5])). The reason is the use of goodness-of-fit tests to assess the goodness of the model chosen. Goodness-of-fit tests can be characterized as unreliable tests as they are very sensitive to the sample

size. Moreover, given that the input distribution is correct, simulation analysts still will not completely know which parameters to use. MLE assists the simulation analysts to estimate the value of the input parameters (e.g., *Exponential*(1.0) vs. *Exponential*(1.5)). This estimation fails to capture the parameter uncertainty in the input distributions and could lead to wrong conclusions from simulation. The validity and credibility of the simulation results and output performance measures depend on the choice of input probability distributions and the associated parameters. The following example illustrates the problem of this practice.

Example 1.1:

Consider the operation of a single-teller bank with an *M*/*M*/1 queue with the arrival rate of λ customers per hour and the service rate is μ customers per hour. The bank opens its doors at 9 A.M. and closes at 5 P.M., but it stays open until all customers at 5 P.M. have been served. Suppose that we have abundant data about the service rate but only few about the arrival rate. With the MLE data fitting, we assume that the resulted $\mu_0 = 10$ is precisely known whereas the resulted $\lambda_0 = 10$ is not. The objective of the single-server queueing simulation is to estimate the expected steady-state sojourn time (time spent in the system y). The bank management would decide if the bank needs to hire a new teller based on the simulation estimation of y. It sees that an acceptable average sojourn time in the system is 0.5 hour per customer at maximum.

In the simulation, the ith customer sojourn time, w_i, is calculated by $w_i = d_i - a_i$, where a_i and d_i are the ith customer's arrival and departure times, respectively. The simulation inputs for the ith customer are its inter-arrival time $\tau_i = a_i - a_{i-1}$ and service time s_i. With the random-number streams $\boldsymbol{u}_1$ for the inter-arrival times and $\boldsymbol{u}_2$ for the

7

service times, we generate $\tau_i = -(1/\lambda)\ln(1 - u_{1i})$ and $s_i = -(1/\mu)\ln(1 - u_{2i})$ by the inverse-transform method (Law, 2007).

The average sojourn time can be estimated as $\hat{t} = (1/C)\sum_{i=1}^{C} w_i$ where the simulation was run long enough to complete the services for C customers. Since the arrival rate is not precisely known with limited data, three possible values $\lambda=\{8, 9, 9.5\}$ are suggested to study the performance of the system. The input parameter λ can be varied to explore different level of imprecision. Table 1-1 shows the output statistics for the three scenarios from $n=10,000$ independent replications of simulation (using Arena® 10.0).

TABLE 1-1
SIMULATION RESULTS FOR N=10,000 INDEPENDENT REPLICATIONS OF THE BANK MODEL IN
ARENA®

λ	Mean: $\hat{y}\,(hrs)$	Half-width: $t_{n-1,1-\alpha/2}\left(s_{\hat{y}}/\sqrt{n}\right)$ ($s_{\hat{y}}$ is sample std. dev.,$\alpha=0.1$)
8	0.3628	<0.01
9	0.4732	<0.01
9.5	0.5432	<0.01

From Table 1-1, we can see that the performance measure sensitivity depends on the input parameter values. If the arrival rate was taken as $\lambda=9.5$, the management would decide to hire a new teller. However, if the estimated arrival rate from input analysis happened to be $\lambda=8$ or $\lambda=9$, the management would not hire a new teller. The parameter estimate largely depends on the *sample size* in the input analysis as well as the time *when* those samples were collected. For this reason, the management could potentially make wrong decisions with a high cost. Besides parameter uncertainty, we may also ask how

reliable the result is with the selection of exponential distributions as the input model. If the independent arrival assumption is not valid, ignoring correlations can have a major effect on the output performance (Livny et al., 1993).

The above simple example shows us the importance of capturing the total uncertainty in simulation. In this example, simulation outputs could lead the management to unintentionally provide a service that is not as good as what is expected or make a high cost decision which is unnecessary.

1.3   Significance of the Problem

Discrete-event simulation has imposed itself as a powerful tool because of its ability to allow its users to imagine their existing and new systems by observing and quantifying the systems' behaviors. With the advancement of computer animation and processing, simulation is used extensively by industry and the government to explore and visualize their systems, and, moreover, to study and compare alternative designs to significant problems. Most of our world decisions are made under uncertainties. Here are some facts to that premise:

- Risk estimation of catastrophic fire incidents in manufacturing facilities, business high-rises, or school buildings may be used to select evacuation plans and suppression routes, where the prior probabilities of such breakouts and costs are uncertain (e.g., Elkins et al., 2007).

- Decisions for design and modification of systems in NASA missions to achieve and maintain high safety standards are made under highly significant uncertainties (e.g., Stamatelatos, 2002).

- The Navy devises strategies to deter terrorist attacks on ports, without knowing attackers' tactics such as the number and speed of attackers (e.g., Lucas et al., 2007).

- Executives design resilient global supply chain networks with the consideration of disruptive natural disasters and political instability, however with limited information about the risk events (e.g., Deleris & Erhun, 2005).

Due to the incomplete descriptions of a mechanism or process and other limitations of scientific knowledge, environmental and ecological problems underlie uncertainties for which risk analysts use different approaches to represent and propagate (e.g., Ferson & Ginzburg, 1996).

Consequently, it becomes more critical to create a reliable simulation mechanism that helps in robust decision making.

1.4   Research Objectives

The objective of this research is to develop a fundamentally different discrete-event simulation framework based on intervals instead of traditional precise numbers. The parameters of probability distributions in simulations are intervals, and probabilities become imprecise. Imprecise probability allows the total uncertainty in simulation to be represented in a concise form, for more details, see Section 2.2. Our goal is to create a new reliable interval-based DES mechanism that accounts for input uncertainties, by executing the following tasks:

1. Input Analysis: methods and algorithms are developed to generate random interval variates from imprecise statistical distribution functions as the simulation input.

10

2. Simulation Mechanism: detailed simulation logic based on intervals including clock advancement and uncertainty propagation is implemented to carry out the simulation.

3. Output Analysis: statistical measures for variability and uncertainty are proposed for interval data interpretation and comparison.

The new simulation framework is to help M&S industries by providing them with a new reliable, understandable, verifiable, easy-to-implement, and efficient DES mechanism that simultaneously incorporates variability and uncertainty by estimating a solution whose completeness and soundness w.r.t. uncertainties can be verified.

1.5   Major Contributions

In this dissertation, we have developed an interval-based discrete-event simulation using probabilistic input distributions with interval parameters. The interval-based formulation of the input distributions models total uncertainty in order to support reliable decision making. The interval-based simulation led us to research the following DES components in an interval-based platform:

1. Parameterization of statistical distributions with interval parameters:

Selecting interval parameters for input distributions in the IBS is based on order-statistics sampling distribution. The objective of the interval parameters is to enclose all possible real-valued scenarios with a certain level of confidence. A relationship is formulated based on order statistics sampling distribution to evaluate the interval parameters which can be used to parameterize single and multiple parameter distributions. In this dissertation, we derive the relationship for the exponential and

normal distributions as examples for single and multiple parameter distributions, respectively.

2. Simulation robustness measure:

As a measure of completeness to support the IBS, we reversely use the derived parameterization relationship to assess the IBS replication length. On the assumption of interval parameters, the required replication length is estimated to enclose the real-valued variates within the interval variates at each order with a certain level of confidence. The relationship is derived in the case of the exponential, normal, and triangular distributions.

3. Interval variate generation:

We propose an interval variate generation technique based on the inverse transform method. Single and multiple parameter distributions are discussed. The exponential distribution is used as an example in regard to the single parameter distributions. In practice, pairs of random variates are generated using the lower and upper parameter values. Applying this on the multiple parameter distribution is not straightforward. All possible combinations of the parameters' values should be considered to generate the enclosing interval variate. Herein, we counter this by studying the interval variate of the normal and triangular distributions.

4. Simulation clock advancement in the IBS:

In this dissertation, we investigate three possible approaches to handle the IBS event list. It is more complex to manage the firing of interval events as they occur within a window of time and their occurrences are not precisely known. It also becomes more difficult when the interval times of the events overlap. Herein, we study three possible approaches to handle the event list of the IBS and the simulation clock advancement

respectively. The three approaches are based on: (1) the lower times to estimate an imprecise best-case scenario, (2) the upper times to estimate an imprecise worst-case scenario w.r.t. performance measure of interest, and (3) a uniformly sampling approach to estimate precise average scenarios that is compared to the Second-order Monte-Carlo (SOMC).

5. Interval-based metamodel for the automated material handling simulation:

We propose an interval-based automated material handling metamodel to simulate a 300m wafer fab. The metamodel is implemented in a Java-based object-oriented simulation package called, *JSim*. *JSim*, a library of Java-based interval DES toolkits, has been developed to support the implementation of the IBS. We modeled two dispatching rules for this application to estimate interval results of the mean response times.

1.6   Thesis Outline

The remainder of this thesis is organized as follows. In Chapter 2, we briefly review the current literature on the input uncertainty quantification methods in DES, and we argue that these methods do not produce complete and sound solutions. Hence, these methods could lead to unreliable conclusions. In order to lay the foundation for the IBS, we survey the imprecise probability representations. The IBS representation of imprecise probability is based on the generalized interval form. We also present Kaucher interval arithmetic and its extension, the generalized intervals as coherent mechanisms to carry out the interval mathematical computations and the design of the interval statistics. We end Chapter 2 by summarizing the research gaps. In Chapter 3, we introduce the interval-based simulation mechanism. The input analysis of our mechanism is discussed by

proposing a standard procedure to determine the interval parameters. In order to assess the validity of the parameterization technique, some SOMC experiments are performed respectively. We also propose a simulation robustness measure regarding the required number of IBS replications to include all possible real-valued scenarios at a certain confidence level. In Chapter 4, we investigate the simulation clock advancement in the IBS. M/M/1 queueing system with interval parameters is employed to illustrate the IBS mechanism using hand simulation. Chapter 4 ends with a discussion about proposed output statistics for interval data. *JSim* is also described as our testbed to run the IBS. Chapter 5 presents an interval-based metamodel approach to estimate the performance of automated materials handling systems as a real-life application of the IBS. In the final chapter, a summary of the dissertation work and outlines for new future research work are given.

CHAPTER 2:
LITERATURE REVIEW

Many methods have been proposed to quantify the input uncertainties in DES. This chapter serves as a literature review of the relevant work dedicated to support reliable DES mechanism and introduces the concepts of imprecise probabilities and interval analysis.

2.1 Input Uncertainty Quantification in Simulation

Input uncertainty in simulation attracted researchers' attentions only recently. The following sections summarize the present state of the uncertainty quantification methods in simulation.

2.1.1 Second Order Monte Carlo Simulation

One of the popular simulation techniques that represent the total uncertainty is the second order Monte Carlo approach (Lan et al., 2007). A second-order probabilistic sensitivity analysis is superimposed on the traditional simulation so that uncertainties are quantified by sampling the parameters of the first-order probabilistic distributions. SOMC contains two simulation loops. The inner loop is the variability loop that reflects the natural variability. The outer loop represents the uncertainty of the input parameters of the inner loop.

SOMC is easy to implement. Yet, the double-loop simulation is computationally costly. In each replication of the outer loop, the simulation output captures one of the possible scenarios associated with the uncertain parameters. As the number of

replications for the outer loop increases, the simulation robustness increases. However, the analyst does not know how many replications to run in order to achieve the desired robustness representing all possible scenarios. The additional question that has to be asked is whether the analyst has enough information to select the distributions of the input parameters in the outer loop.

Further, the completeness of response measurement is not easily verified in SOMC. In SOMC, the soundness of the response measurement is guaranteed if the outer loop distributions for the input parameters are valid. However, the completeness is not verifiable unless the number of replications for the outer loop increases tremendously.

2.1.2   Bayesian Methods

The basic idea of the Bayesian analysis for input uncertainty in simulation is to place a prior distribution on each input parameter in simulation to describe its initial uncertainty. The prior distribution is then updated to a posterior distribution based on the observed data. The two distributions are used to reduce uncertainty about the parameters.

Glynn (1986) first proposed a general Bayesian approach to continuously update input distributions with real-world data. Chick (1997) suggested the applications of Bayesian analysis to a broader range of areas such as input uncertainties, rankings, response surface modeling, and experimental design. Andradóttir and Bier (2000) also proposed the Bayesian analysis approach for input uncertainties and model validation. A Bayesian model average (BMA) method, developed by Chick (1999, 2000, 2001), is used when multiple candidate distributions are proposed for a single source of randomness. It estimates the model and input parameters based on posterior distributions. The algorithm

generates independent and identically distributed (IID) replicates of the estimates by sampling a single model and parameter from the posterior. The parameters of the chosen model are then updated. This process is repeated and the output is averaged (Ng & Chick, 2006). The idea of BMA was further improved by Zouaoui and Wilson (2001a, 2001b, 2003, 2004). In their new version of BMA-based simulation algorithm, the analyst has more control on the number of simulation replications to be performed.

The Bayesian methods quantify the parameter uncertainty in the simulation response. However, the difficulty of computing posterior distributions hindered the wide spread of this method. In practice, the analyst needs more computational procedures such as Markov chain Monte Carlo simulation or importance sampling to implement this method. The non-generality of the methods has also reduced their use since they need to be tailored to each application (Henderson, 2003).

2.1.3   Delta Method

Cheng and Holland proposed a Delta method (Cheng, 1994; Cheng & Holland, 1997, 1998), and the framework was also adopted by Zouaoui and Wilson (2001a, 2001b). The framework of this method assumes that the model is known while input parameters are uncertain. The true values of the parameters are estimated using the MLE assuming that the parameters follow a normal distribution. This estimation is valid under mild regularity conditions.

The total output variance of simulation is estimated by two terms. The first term is the simulation variance, and the second term is the input parameter variance. The early work of Cheng and Holland (Cheng, 1994; Cheng & Holland, 1997, 1998) did not

17

include bias in the mean square error (MSE) of the parameters. The failure of including bias is substantial in the sense that the simulation output confidence intervals are conservative. Hence, the variance is overestimated (Henderson, 2003). The improved method (Cheng & Holland, 2004) considers the bias in the MSE of the parameters, which also needs less computational effort.

Nonetheless, its major disadvantage is in the assumption that the model is known with certainty. Furthermore the performance of this method is not yet known compared to the other methods such as Bayesian and Bootstrap methods (Henderson, 2003).

2.1.4   Bootstrap Approach

Barton and Schruben (2001) proposed three non-parametric resampling methods to incorporate the error due to input distributions. These methods use empirical distribution functions (EDFs) to model the distribution functions of independent input random variables. For parametric resampling, Cheng and Holland (1997, 2004) quantified the effect of parameter uncertainty for the parametric formulation. With new observations in bootstrap, estimates of input parameters are continuously updated using the MLE. From each bootstrap, a simulation experiment is conducted to give a simulation average output. A percentile confidence interval of the simulation output can be calculated.

The use of percentile confidence interval in bootstrapping methods assumes the absence of simulation uncertainty or variability. When simulation uncertainty is present, percentile confidence intervals are based on a convolution of the input uncertainty and

simulation uncertainty. Hence, it is not clear how these intervals behave (Henderson, 2003).

The reliability of our interval-based simulation mechanism stands on the imprecise probability theory, which is introduced in Section 2.2.

2.2 Imprecise Probability

Instead of a precise value of the probability $P(E) = p$ associated with an event E, a pair of lower and upper probabilities $P(E) = [\underline{p}, \overline{p}]$ is used to include a *set* of probabilities and quantify the uncertainty. Imprecise probability differentiates uncertainty from variability both qualitatively and quantitatively, which is the alternative to the traditional sensitivity analysis in probabilistic reasoning to model indeterminacy and imprecision.

Many representations of imprecise probabilities have been proposed. For example, the Dempster-Shafer evidence theory (Dempster, 1967 & Shafer, 1990) characterizes evidence with discrete probability masses associated with a power set of values, where Belief-Plausibility pairs are used to measure uncertainties. The behavioral imprecise probability theory (Walley, 1991) models uncertainties with the lower prevision (supremum acceptable buying price) and the upper prevision (infimum acceptable selling price) with behavioral interpretations. The possibility theory (Dubois & Prade, 1988) represents uncertainties with Necessity-Possibility pairs. A random set (Malchanov, 2005) is a multi-valued mapping from the probability space to the value space. Probability bound analysis (Ferson et al., 2002) captures uncertain information with p-boxes which are pairs of lower and upper distribution functions. F-probability

(Weichselberger, 2000) incorporates intervals into probability values which maintain the Kolmogorov properties. Fuzzy probability (Möller & Beer, 2004) considers probability distributions with fuzzy parameters. A cloud (Neumaier, 2004) is a combination of fuzzy sets, intervals, and probability distributions. Recently, an imprecise probability with a generalized interval form (Wang, 2008a, 2008b) was also proposed, where the probabilistic calculus structure is simplified based on the algebraic properties of the Kaucher arithmetic (Kaucher, 1980) for generalized intervals.

Imprecise probability captures the total uncertainty and represents its two components quantitatively. It can provide a concise form to improve the robustness of simulation without the traditional sensitivity analysis related procedures. Interval-valued imprecise probabilities can help to simulate a set of scenarios for each simulation run. Interval arithmetic provides the calculus structure, models uncertainty propagation, and ensures the completeness of range estimation, as introduced in Section 2.3.

2.3 Interval Analysis

2.3.1 Generalized Intervals

Interval mathematics (Moore, 1966) is a generalization in which interval numbers replace real numbers, interval arithmetic replaces real arithmetic, and interval analysis replaces real analysis. Interval arithmetic was originally developed to solve the issue of numerical errors in digital computation due to the floating-point representation of numbers, where rounding and cancellation errors put the reliability of digital computation at risk. Not only do intervals solve the problem of representation for real numbers on a

digital scale, but they also provide a generic form to represent uncertainties and errors in technical construction, measuring, computation, and range of fluctuation.

Interval arithmetic considers all possibilities of variation even in the worst cases of uncertainty propagation. Let $[\underline{x}, \overline{x}]$ and $[\underline{y}, \overline{y}]$ be two real intervals (i.e., $\underline{x}, \overline{x}, \underline{y}, \overline{y} \epsilon R$) and $\circ$ be one of the four basic arithmetic operations for real numbers $R, \circ \in \{+, -, \times, \div\}$. The set-based enclosure for intervals $[\underline{x}, \overline{x}]$ and $[\underline{y}, \overline{y}]$ is $[\underline{x}, \overline{x}] \circ [\underline{y}, \overline{y}] = \{x \circ y | x \in [\underline{x}, \overline{x}], y \in [\underline{y}, \overline{y}]\}$. The corresponding interval arithmetic operations are defined for the worst cases. For example, $[\underline{x}, \overline{x}] + [\underline{y}, \overline{y}] = [\underline{x} + \underline{y}, \overline{x} + \overline{y}], [\underline{x}, \overline{x}] - [\underline{y}, \overline{y}] = [\underline{x} - \overline{y}, \overline{x} - \underline{y}]$ and $[\underline{x}, \overline{x}] \times [\underline{y}, \overline{y}] = [\min(\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}), \max(\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y})]$. When the lower and upper bounds of an interval are equal, the point-wise interval is the same as a real number.

In interval arithmetic, it is guaranteed that intervals calculated from arithmetic include all possible combinations of real values within the respective input intervals. That is, $\forall x \in [\underline{x}, \overline{x}], \forall y \in [\underline{y}, \overline{y}], \exists z \; x \in [\underline{x}, \overline{x}] \circ [\underline{y}, \overline{y}], x \circ y = z$. For example, $[1,3] + [2,4] = [3,7]$ guarantees that $\forall x \in [1,3], \forall y \in [2,4], \exists z \; [3,7], x + y = z$. Similarly, $[3,7] - [1,3] = [0,6]$ guarantees that $\forall x \in [3,7], \forall y \in [1,3], \exists z \in [0,6], x - y = z$. This is an important property that ensures the completeness of range estimations. When input variables are not independent, the output results will over-estimate the actual ranges. This only affects the soundness of estimations, not their completeness. Some special techniques have also been developed to avoid the range over-estimations based on monotonicity properties of interval functions.

Generalized interval (Gardenes et al., 2001) is an extension of the above set-based classical interval with better algebraic and semantic properties based on the Kaucher arithmetic (Kaucher, 1980). A generalized interval $[\underline{x}, \overline{x}]$ is not constrained by $\underline{x} \le \overline{x}$ any more. Therefore, $[4,2]$ is also a valid interval and called improper, whereas the traditional interval $[\underline{x}, \overline{x}]$ with $\underline{x} \le \overline{x}$ is called proper. The relationship between proper and improper intervals is established with the operator dual. Given a generalized interval $\mathbf{x} = [\overline{x}, \underline{x}]$, then $dual\mathbf{x} = [\underline{x}, \overline{x}]$.. Based on the Theorems of Interpretability (Gardenes et al., 2001), generalized interval provides more semantic power to help verify completeness and soundness of range estimations by logic interpretations. The four examples in Table 2-1 illustrate the interpretations for operator "+", where the range estimation of $[\underline{z}, \overline{z}] = [4,7]$ in the 1st row is complete and the estimation of $[\underline{z}, \overline{z}] = [7,4]$ in the 4th row is sound. $-, \times, /$ have the similar semantic properties. More information about generalized intervals can be found in (Wang, 2008b, 2008c, 2008d).

TABLE 2-1
ILLUSTRATIONS OF THE SEMANTIC EXTENSION OF GENERALIZED INTERVAL

Algebraic Relation: $[\underline{x}, \overline{x}] + [\underline{y}, \overline{y}] = [\underline{z}, \overline{z}]]$	Corresponding Logic Interpretation	Quantifier of $[\underline{z}, \overline{z}]]$	Estimation of $[\underline{z}, \overline{z}]]$
$[2,3] + [2,4] = [4,7]$	$(\forall x \in [2,3])(\forall y \in [2,4])(\exists\, z \in [4,7])(\,x + y = z)$	$\exists$	complete
$[2,3] + [4,2] = [6,5]$	$(\forall x \in [2,3])(\forall z \in [5,6])(\exists\, y \in [2,4])(\,x + y = z)$	$\forall$	sound
$[3,2] + [2,4] = [5,6]$	$(\forall y \in [2,4])(\forall x \in [2,3])(\exists\, z \in [5,6])(\,x + y = z)$	$\exists$	complete
$[3,2] + [4,2] = [7,4]$	$(\forall z \in [4,7])(\forall x \in [2,3])(\exists y \in [2,4])(\,x + y = z)$	$\forall$	sound

In our new simulation mechanism, uncertainty propagation will be modeled based on both the interval arithmetic and Kaucher arithmetic. This allows us to interpret interval results so that the completeness and soundness can be verified rigorously. Since the simulation performance measures are also intervals, statistics based on intervals should be used to draw conclusions from the simulation results. The performance statistics that have been recently studied are mean and variance, as summarized in the following section.

2.3.2   Interval Statistics

The mean of a set of random intervals $\{[\underline{x_i}, \overline{x_i}] | \underline{x_i} \leq \overline{x_i}, \underline{x_i} \in \mathbb{R}, \overline{x_i} \in \mathbb{R}\}$ where $i = 1, \dots, N$ is also an interval. It should include the smallest possible and the largest possible means which can be calculated from any possible enclosed real number $x_i \in [\underline{x_i}, \overline{x_i}]$. Because the formula to calculate the mean is a monotone function, the lower bound of the interval mean is just the average of the left endpoints $\underline{x_i}$'s, and the upper bound is the average of the right endpoints $\overline{x_i}$'s (Granvilliers et al., 2003). Therefore, the *arithmetic mean* of random intervals is given by

$$\left[\underline{\mu}, \overline{\mu}\right] = \left[\frac{1}{N}\sum_{i=1}^{N} \underline{x_i}, \frac{1}{N}\sum_{i=1}^{N} \overline{x_i}\right] \tag{1}$$

where N is the sample size of the random intervals.

Computing the range for the variance $\left[\underline{V}, \overline{V}\right]$ for a set of intervals is an NP-hard problem (Granvilliers et al., 2003). Several algorithms (Ferson et al., 2007; Granvilliers et al., 2003; Xiang et al., 2007a) were proposed to obtain the bounds of the variance. It

was found that $\underline{V}$ can be computed in $O(N\log N)$ computational steps for N interval data samples. However, computing the upper bound of the variance $\overline{V}$ requires the computational effort that grows exponentially with the number of intervals in the data set. Only for several special cases, when intervals do not overlap and there is no interval completely nested in another, $O(N\log N)$ and linear time algorithms are available to compute $\overline{V}$.

In this dissertation, we propose new measures of interval statistics for output interpretation in reliable simulation to support decision makings. Compared to the variance measures mentioned above, our measures are much easier to compute thus more applicable in large-scale simulations.

2.4 Discussion of Research Gaps

After reviewing the state of the art of the uncertainty quantification in simulation methods, we observed the following research gaps:

1. The completeness and soundness of simulation measures for uncertainties are not verifiable. Hence, the robustness of the simulation output is not guaranteed.

2. The computation is expensive and implementations are complex for simulation practitioners.

Our proposed mechanism is to solve these two issues because it incorporates variability and uncertainty components based on imprecise probabilities. The new mechanism does not require enormous computational procedures (e.g. Markov chain Monte Carlo simulation). On the contrary, it requires less computational effort because an

IBS replication produces a solution range in an interval form instead of a single real-valued number.

CHAPTER 3:
IBS METHDOLOGY AND INPUT ANALYSIS

This chapter introduces the concept of the interval-based simulation mechanism. Section 3.1 covers the theoretical aspects of the mechanism. The simulation aspects of the input analysis and the uncertainty propagation in the IBS are discussed in Sections 3.2 and 3.3, respectively.

3.1   The Proposed Interval-based Simulation

In this research, we propose a reliable interval-based simulation mechanism to account for input uncertainties. The new simulation models are based on intervals instead of real numbers in order to help obtain more reliable estimates of outputs. Interval-valued imprecise probabilities are used, and interval random variates are generated for simulation. For each run, intervals as ranges of possibilities are given as output performance measures.

For instance, in the example of Section 1.2, we model the inter-arrival and service times by two exponential distributions with interval parameters. Figure 3-1 illustrates the modeling of the interval-based simulation mechanism to the bank example mentioned in Section 1.2. We use the notation [M]/[M]/1 to represent a single server with inter-arrival times that are exponentially distributed with interval parameter, i.e. $\exp\left(\left[\underline{\lambda}, \overline{\lambda}\right]\right)$ and service times that are also exponentially distributed, i.e. $\exp\left(\left[\underline{\mu}, \overline{\mu}\right]\right)$. From the imprecise probability distributions, *random intervals* as uncertain random variates are generated, such as arrival time $[\underline{a}_i, \overline{a}_i]$ and service time $[\underline{s}_i, \overline{s}_i]$.
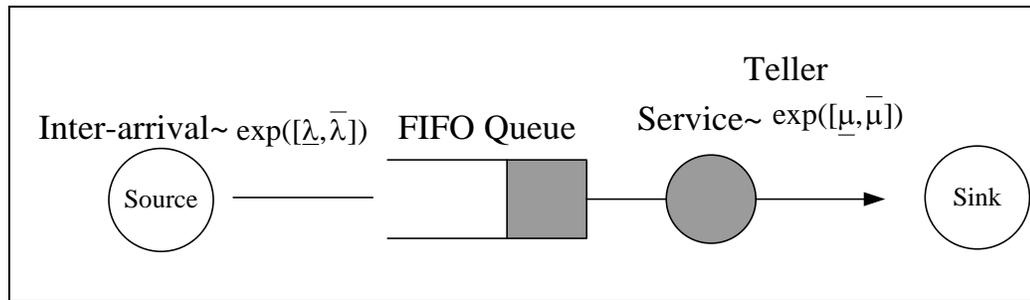
Figure 3-1: Bank example with [M]/[M]/1 queue

This representation captures *both* parameter and model uncertainties, since one interval-valued distribution actually models a ***set*** of distributions simultaneously. Consequently, the IBS is described by a pair of cumulative distributions functions (*cdf's*) corresponding to the lower and upper bounds, instead of a crispy *cdf* obtained from traditional simulations. In literature, the lower and the upper bounds are referred to as p-box, (Ferson & Donald, 1998). Figure 3-2 shows the upper and lower cumulative distribution functions obtained from an interval-based simulation multiple real-valued *cdf's*.
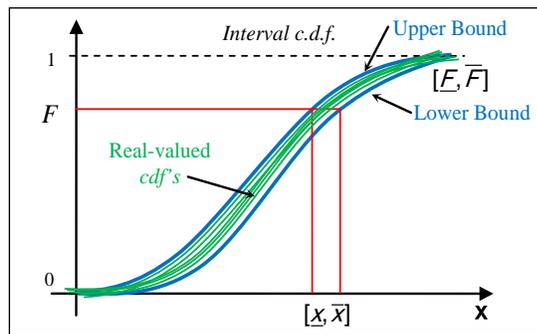
Figure 3-2: Upper and lower bounds from an interval-based simulation enclosing multiple *cdf's*

Interval-based simulation is much more efficient than the traditional Second-Order Monte-Carlo sampling approaches based on real numbers, where many runs are needed to estimate a similar range. Technically, if a *complete* range estimation between a and b is required from a bounded distribution such as the uniform $U(a, b)$, the sample size should be large enough such that the random numbers from a *full-period* pseudo-random number generator are exhausted, which is in the scale of 10^{18} or more. Therefore, the interval-based simulation is also more reliable than real-valued estimates with the same number of runs.

Random intervals represent the two components of total uncertainty in simulation. Here, probabilistic distributions represent variabilities, and intervals capture uncertainties. With this interval representation of parameters, the degree of uncertainties is captured by the intervals' widths. The larger the parameter interval width is, the less knowledge we have about this parameter, and vice versa. The goal of our reliable simulation mechanism is to incorporate input uncertainties in simulation and provide decision makers with timely and comprehensive insights of complex systems in order to make robust decisions.

3.2 Input Analysis

To carry out the IBS using interval random inputs, we have the following three major input analysis tasks:

1. Finding the probability distributions of the random inputs with interval parameters is discussed in Section 3.2.1.

2. Generating interval random variates from the distribution functions, as shown in Section 3.2.2.

3. Designing a simulation robustness measure that quantifies the confidence we have that a simulation result based on intervals would include all possible scenarios if real-valued *cdf*'s are used as in the traditional simulation, as illustrated in Section 3.2.3.

3.2.1 Hypothesizing and Parameterization of Interval Probability Distributions

An important question is how to select the probability distributions and interval parameters. In section 3.2.1.1, we propose a standard procedure to determine interval parameters of probability distributions when data is available. Section 3.2.1.2 addresses the question in the absence of data.

3.2.1.1 In the Presence of Data

If it is possible to collect data for an input random variate of interest, the set of data is used to fit a theoretical interval based distribution form. First, the data is used to build a theoretical distribution with real-valued parameters in the traditional approaches. In simulation applications, it is common that MLE guides the selection of the distributions parameters. Based on the obtained distribution with real-valued parameters and the replication length n that the analyst can afford to run, the lower and the upper bounds of the interval parameters are estimated. The parameter bounds are calculated such that all possible real-valued scenarios are included in the simulation output with a certain confidence level of $(1-\alpha)$.

This confidence is interpreted as the probability of having an assumed real-valued random variable x bounded by the corresponding random interval $[\underline{x}, \overline{x}]$ at any cumulative probability p in *cdf*. The goal is to achieve the probability of enclosure

$$P(x \in [\underline{x}, \overline{x}]) \geq 1 - \alpha$$

(2)

where x is the random variable if the simulation is run from any real-valued parameter bounded by the interval parameter. Extending the notation of probability in Eq.(2), we can write it as follows assuming the independence of the lower and upper bounds

$$P(\underline{x} \leq x \leq \overline{x}) = P(\underline{x} \leq x) \times (1 - P(\overline{x} \leq x))$$

(3)

Order statistics sampling distribution is used to ensure that the probability in Eq.(3) at any cumulative probability p is at least $(1 - \alpha)$. If the real-valued variables are ordered as $x_{(1)}, x_{(2)}, \dots x_{(n)}$, the corresponding value of p associated with the rth ordered observation is given by $(r - 0.5)/n$. The sampling distribution of the transformed order statistics *cdf* is given by $G_r(x)$. $G_r(x)$ is interpreted as the probability that at least r observations in the sample do not exceed x and can be calculated as (Stuart, 1987)

$$G_r(x) = \sum_{j=r}^{n} \left[\binom{n}{j} (F(x))^j (1 - F(x))^{n-j} \right]$$

(4)

where $F(x)$ is the *cdf* of the random variable x.

Based on the ordered statistics sampling distribution, the probability of having the rth random variable $x_{(r)}$ between the rth bounds of the interval random variable $[\underline{x}_r, \overline{x}_r]$ is given by

$$P\left(\underline{x}_r \leq x_r \leq \overline{x}_r\right) = \underline{G}(x_r) \times \left(1 - \overline{G}(x_r)\right)$$

$$(5)$$

where $\underline{G}(x_r)$ and $\overline{G}(x_r)$ are the upper and the lower sampling distribution, respectively.

To find the lower parameter interval limit, we set the upper sampling distribution $\underline{G}(x_r)$ at any order r to (1-α) as

$$\underline{G}(x_r) = \sum_{j=r}^{n}\left[\binom{n}{j}\left(\underline{F}(x_r)\right)^{j}\left(1 - \underline{F}(x_r)\right)^{n-j}\right] \geq 1 - \alpha$$

$$(6)$$

where x_r is calculated from the inverse transform of the assumed distribution function with real-valued parameters as

$$x_r = F^{-1}((r - 0.5)/n)$$

$$(7)$$

The probability in Eq.(6) can be used for any probabilistic distribution function by replacing the upper cumulative distribution function $\underline{F}(x)$ with the corresponding distribution form. The lower interval parameter limit is first assumed as the real-valued parameter and then it is decreased gradually until the desired probability of (1-α) is achieved.

On the other hand, for the upper interval parameter limit, we set the complement of the lower sampling distribution $\left(1 - \overline{G}_r(x)\right)$ at any order r to (1-α) as

$$\left(1 - \overline{G}(x_r)\right) = \left(1 - \sum_{j=r}^{n}\left[\binom{n}{j}\left(\overline{F}(x_r)\right)^{j}\left(1 - \overline{F}(x_r)\right)^{n-j}\right]\right) \geq 1 - \alpha$$

$$(8)$$

The probability in Eq.(8) can be used for any probability distribution function by replacing the lower cumulative distribution function $\overline{F}(x)$ with the corresponding distribution form. The upper interval parameter is obtained by increasing its value until

the probability of (1-α) is achieved. The following illustrates the interval parameterization technique for single and multiple parameter distributions.

Single Parameter Distribution

As an example, we demonstrate the interval parameterization for single parameter distribution with the exponential distribution. Here we derive the specific form of Eq.(6) and Eq.(8) for the exponential distribution. Assume a stochastic process follows an exponential distribution with an estimated real-valued rate of β. An interval exponential distribution with the rate of $\left[\underline{\beta}, \overline{\beta}\right]$ is used to enclose the real-valued *cdf*, where $\beta \epsilon [\underline{\beta}, \overline{\beta}]$. The upper bound *cdf* is associated with $\underline{\beta}$ and the lower bound *cdf* is with $\overline{\beta}$. Substituting the exponential upper cumulative distribution function $\underline{F}(x_r) = 1 - e^{-x_r/\underline{\beta}}$ and the random variate $x_r = -\beta \ln\left(1 - (r - 0.5)/n\right)$ at order r in Eq.(6), we receive

$$P(\underline{x}_r \leq x_r) = \sum_{j=r}^{n}\left[\binom{n}{j}\left(1 - \left(1 - \frac{r - 0.5}{n}\right)^{\beta/\underline{\beta}}\right)^{j} \times \left(\left(1 - \frac{r - 0.5}{n}\right)^{\beta/\underline{\beta}}\right)^{n-j}\right] \tag{9}$$

With the exponential lower cumulative distribution function $\overline{F}(x) = 1 - e^{-x/\overline{\beta}}$ and the random variate $x_r = -\beta \ln\left(1 - (r - 0.5)/n\right)$ at order r in Eq.(8), we receive

$$P(x_r \leq \overline{x}_r) = 1 - \sum_{j=r}^{n}\left[\binom{n}{j}\left(1 - \left(1 - \frac{r - 0.5}{n}\right)^{\beta/\overline{\beta}}\right)^{j} \times \left(\left(1 - \frac{r - 0.5}{n}\right)^{\beta/\overline{\beta}}\right)^{n-j}\right] \tag{10}$$

The lower interval mean $\underline{\beta}$ at any order r is calculated as follows. Given a particular value of β and the available sample size n, set $\underline{\beta} = \beta$, then gradually reduce the value of $\underline{\beta}$ to compute the probability of enclosure using Eq.(9) until it reaches the predetermined probability of $(1 - \alpha)$. The resulted $\underline{\beta}$ is the value satisfying the desired probability at the predetermined order r. Similarly, use Eq.(10) to find $\overline{\beta}$ for the desired

probability of enclosure by gradually increasing the initial value of $\overline{\beta} = \beta$. Since the

parameter is sensitive up to three significant digits, 0.001 is used as the incremental step

size. Figure 3-4 illustrates the computation algorithm for the exponential distribution.

To find the lower interval parameter $\underline{\beta}$ at any order r:	To find the upper interval parameter $\overline{\beta}$ at any order r:
Step 0: Given β, n, and α	*Step 0:* Given β, n, and α
Step 1: Set $\underline{\beta} = \beta$	*Step 1:* Set $\overline{\beta} = \beta$
Step 2: Calculate the probability p in Eq.(9)	*Step 2:* Calculate the probability p in Eq.(10)
Step 3: If $(p \geq 1 - \alpha)$	*Step 3:* If $(p \geq 1 - \alpha)$
return $\underline{\beta}$	return $\overline{\beta}$
Else $\underline{\beta} = \underline{\beta} - 0.001$,	Else $\overline{\beta} = \overline{\beta} + 0.001$,
Back to *Step 2.*	Back to *Step 2.*

Figure 3-3: The algorithm to calculate the interval parameter for an exponential
distribution

The probability of enclosure in Eq.(9) and Eq.(10) can be verified by SOMC

simulation (Batarseh & Wang, 2009). The following numerical example is used to

illustrate.

Example 3.1: For an exponential distribution with a rate of $1/\beta$ where β is assumed to

follow a uniform distribution U[0.111,0.143]. For each sampled mean β from its uniform

distribution, the corresponding $\underline{\beta}$ and $\overline{\beta}$ are calculated based on the algorithm in Figure 3-

4 at three orders $r = 250, r = 500,$ and $r = 750$ at a confidence level of $\alpha = 0.1$. Table 3-1

shows the results for three orders. The tabulated values of $\underline{\beta}$ is the minimum value of the

calculated $\underline{\beta}$'s based on Eq.(9) at a particular order, whereas the tabulated $\overline{\beta}$ is the

maximum value of the calculated $\overline{\beta}$'s based on Eq.(10). The interpretation of those

intervals is that the resulted bounds from a single IBS replication with a length of n using

the calculated interval parameter at a particular order will enclose at least 90% of the

SOMC *cdf*'s.

TABLE 3-1
INTERVAL MEAN FOR AN EXPONENTIAL DISTRIBUTED PROCESS AT $r = 250, r = 500, and\ r = 750$

r	250	500	750
$[\underline{\beta}, \overline{\beta}]$	[0.102,0.155]	[0.105,0.151]	[0.106,0.150]

The enclosure at the desired probability of at least 90% is verified by the

following experiments. We ran SOMC where $\beta \sim U[0.111, 0.143]$ for one thousand

replications, i.e. the outer loop was run $n = 1000$ times. The IBS was run a single

replication with a length of n at a particular order using the corresponding interval mean

from Table 3-1. Table 3-2 shows the obtained probability of enclosing the 1000 SOMC

replications between the lower and upper bounds of the IBS for the three orders.

TABLE 3-2
PROBABILITY OF ENCLOSING 1000 SOMC REPLICATIONS BETWEEN THE LOWER AND THE
UPPER BOUNDS AT $r = 250, r = 500$, AND $r = 750$

r	250	500	750
$P(\underline{x}_r \le x_r \le \overline{x}_r)$	91%	96.2%	91.5%

The probabilities in Table 3-2 are above 90%, which is the desired confidence

level. From Table 3-2, we can notice that the probability at the small and the high orders

represented by $r = 250$ and $r = 750$, respectively, are very close to 90% compared to the middle order represented by $r = 500$. This is due to the narrow width of the *cdf* bounds at the low and the high cumulative probability. Figure 3-4 shows an example of the desired enclosure for five SOMC replications between the lower and the upper bounds of the IBS. With regard to the simulation time, the IBS replication needed 1 wall-time second to be run while the 1000 SOMC replications required 158 wall-time seconds. The IBS has offered a saving of 99.4% of the simulation time with a confidence of enclosing at least 90% of the SOMC replications.
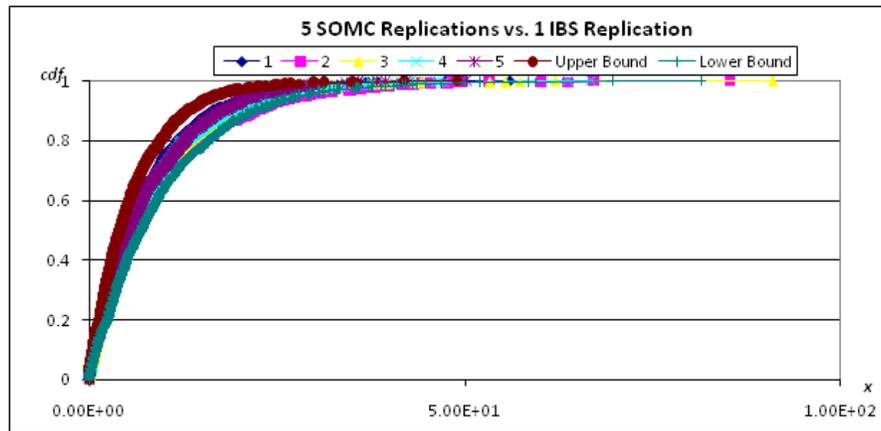
Figure 3-4: Example for the enclosure of five SOMC replications between the bounds of IBS for the exponential distribution in Example 3.1

Multiple Parameter Distributions

If a candidate family of distributions with multiple parameters is hypothesized using MLE, we must somehow specify the values of their interval parameters in order to specify the variables distributions and use them in the IBS.

Normal Distribution

For instance, assume that X is a random variable, and simulation analysts hypothesize that the underlying distribution of X is a normal distribution with a mean μ and a variance of σ^2, which can be written as $X \sim N(\mu, \sigma^2)$.

However, process owners can only characterize whether the parameters are precisely known or not. We assume that both parameters are imprecise. In this case, the distribution of X can be extended to $X \sim N\left(\left[\underline{\mu}, \overline{\mu}\right], \left[\underline{\sigma^2}, \overline{\sigma^2}\right]\right)$. One possible approach to build the lower and the upper bounds of the normal distribution is based on the confidence intervals (Aughenbaugh & Paredis, 2005). The confidence intervals of the mean and the standard deviation for a normal distribution can be estimated (Mood & Graybill, 1963). According to the Central Limit Theorem, the mean is normally distributed with the same mean μ and a variance of $\frac{\sigma^2}{n}$. Therefore, the confidence interval of the mean is calculated as

$$\left[\underline{\mu}, \overline{\mu}\right] = \left[\mu - \frac{t_{\frac{\alpha}{2}, n-1}\,\sigma}{\sqrt{n}}, \mu + \frac{t_{\frac{\alpha}{2}, n-1}\,\sigma}{\sqrt{n}}\right] \tag{11}$$

where t is a quantity obtained from Student distribution with $(n-1)$ degrees of freedom at a cumulative probability $\left(1 - \frac{\alpha}{2}\right)$, and n is the sample size of the available data. The interval in Eq.(11) means that each time we use the resulted confidence intervals to estimate the mean, the intervals contain the true value of the mean $(1 - \alpha)$ number of times. In addition, the variance of the normal distribution follows a chi-square distribution with $(n-1)$ degrees of freedom. The confidence interval for the variance can be estimated as

$$[\underline{\sigma}^2, \overline{\sigma}^2] = \left[\frac{(n-1)s^2}{\chi^2_{\frac{\alpha}{2}, n-1}}, \frac{(n-1)s^2}{\chi^2_{1-\frac{\alpha}{2}, n-1}} \right] \tag{12}$$

where $\chi^2_{a,b}$ is a quantity obtained from the chi-distribution at a confidence level a and b degrees of freedom. The intervals in Eqs.(11) and (12) guarantee that the mean and the variance are enclosed $(1-\alpha)$ of the times between the bounds of their estimated intervals. However, this enclosure of the parameters does not guarantee the enclosure of the real-valued *cdf* between the lower and the upper *cdf*'s generated from these intervals.

The general motivation of the interval parameterization in the IBS is to enclose the real-valued *cdf* between the lower and the upper *cdf*'s with a certain level of confidence as shown in Eq.(5). This enclosure of the input distributions gives interval simulation results that contain the real-valued results with a certain level of confidence. Therefore, we use the order statistics sampling distribution to estimate the interval parameters of the normal distribution.

Order statistics sampling distribution can be used to quantify the parameters' bounds at any order r. However, there are two parameters, namely μ and σ, for which we need to estimate the bounds. There are four combinations of μ and σ for a random variable $X \sim N\left(\left[\underline{\mu}, \overline{\mu} \right], \left[\underline{\sigma}^2, \overline{\sigma}^2 \right] \right)$, as illustrated in Figure 3-5. The combinations are formed as follows: $c_1 \sim N(\underline{\mu}, \overline{\sigma}^2)$, $c_2 \sim N(\underline{\mu}, \underline{\sigma}^2)$, $c_3 \sim N(\overline{\mu}, \underline{\sigma}^2)$, $c_4 \sim N(\overline{\mu}, \overline{\sigma}^2)$. Figure 3-5 shows that the $2^2 = 4$ combinations intersect at order $cdf = 0.5$. In addition, the four combinations form the extreme boundaries that compose the lower and the upper bounds. The combinations are arranged increasingly in a different manner for a different *cdf* illustrated in Figure 3-5.
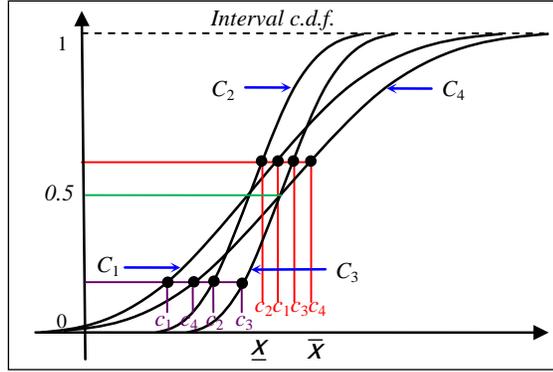
Figure 3-5: Four combinations for the normal distribution with interval parameters

Order statistics sampling distribution is used to find the interval mean $\left[\underline{\mu}, \overline{\mu}\right]$ at any order. First, we estimate the confidence interval of variance using the chi-distribution as in Eq.(12). Herein, we derive the lower and upper sampling distribution for a normally distrusted random variable, as in Eq.(13) and Eq.(14), respectively.

The lower sampling distribution for a normally distributed process is given as

$$P\left(\underline{x}_r \leq x_r\right) = \left(\sum_{j=r}^{n}\left[\binom{n}{j}\left(\frac{1}{2}+\frac{1}{2}erf\left(\frac{x_r-\underline{\underline{\mu}}}{\sigma_l/\sqrt{2}}\right)\right)^j\times\left(\frac{1}{2}-\frac{1}{2}erf\left(\frac{x_r-\underline{\underline{\mu}}}{\sigma_l/\sqrt{2}}\right)\right)^{n-j}\right]\right)$$

(13)

where

$$\sigma_l = \begin{cases}\overline{\sigma} & (r < n/2) \\ \underline{\sigma} & (r > n/2)\end{cases}$$

and the error function is given as

$$erf(x) = 2/\sqrt{\pi}\int_0^x e^{-t^2}\,dt$$

The real-valued variate at order r is estimated as follows

$$x_r = \mu + \sqrt{2}\sigma\,erf^{-1}\left(\frac{2r-1}{n}-1\right)$$

38

and erf^{-1} is the inverse error function. The lower bound is generated from the lower mean $\underline{\mu}$ at any order r from Eq.(13). However, if order r is less than $\frac{n}{2}$ then the lower bound is generated from the upper variance $\overline{\sigma}^2$, and it switches to the lower variance $\underline{\sigma}^2$ if r is greater than $\frac{n}{2}$. The upper sampling distribution is given as

$$P(\overline{x}_r \geq x_r) = \left[1 - \left(\sum_{j=r}^{n} \left[\binom{n}{j} \left(\frac{1}{2} + \frac{1}{2} erf \left(\frac{x_r - \overline{\mu}}{\sigma_u / \sqrt{2}} \right) \right)^j \times \left(\frac{1}{2} - \frac{1}{2} erf \left(\frac{x_r - \overline{\mu}}{\sigma_u / \sqrt{2}} \right) \right)^{n-j} \right] \right) \right] \tag{14}$$

where

$$\sigma_u = \begin{cases} \underline{\sigma} & (r < n/2) \\ \overline{\sigma} & (r > n/2) \end{cases}$$

Similarly, the upper bound is generated from the upper mean at any order r from Eq.(14). The switching of the variance occurs at $cdf = 0.5$. The lower variance $\underline{\sigma}^2$ generates the upper bound if order r has a $cdf < 0.5$. On the other hand, the upper bound is generated from the upper variance $\overline{\sigma}^2$ for orders that have a $cdf > 0.5$. The difficulty of the normal distribution enclosure occurs at the middle orders. The bounds become wider at small and large orders. Thus, wider interval means are obtained at the middle orders to enclose the real variates and the means get narrower at the small and large orders for the same confidence level of enclosure. The algorithms to generate the lower and upper bounds of means are shown in Figure 3-6.

To find the lower interval parameter $\underline{\mu}$ at any order r:	To find the upper interval parameter $\overline{\mu}$ at any order r:
Step 0: Given μ, $[\underline{\sigma},\overline{\sigma}]$, n, and α	**Step 0:** Given μ, $[\underline{\sigma},\overline{\sigma}]$, n, and α
Step 1: Set $\underline{\mu} = \mu$	**Step 1:** Set $\overline{\mu} = \mu$
Step 2: Calculate the probability p in Eq.(13), $$\text{if } r < \frac{n}{2}, \sigma = \overline{\sigma},$$ $$\text{else } \sigma = \underline{\sigma}.$$	**Step 2:** Calculate the probability p in Eq.(14) $$\text{if } r < \frac{n}{2}, \sigma = \overline{\sigma},$$ $$\text{else } \sigma = \underline{\sigma}.$$
Step 3: If $(p \geq 1 - \alpha)$ return $\underline{\mu}$, Else $\underline{\mu} = \underline{\mu} - 0.001$, Back to *Step 2*.	**Step 3:** If $(p \geq 1 - \alpha)$ return $\overline{\mu}$, Else $\overline{\mu} = \overline{\mu} + 0.001$, Back to *Step 2*.

Figure 3-6: The algorithm to calculate the interval parameters for a normal distribution

Triangular Distribution

Assume that the candidate distribution follows a triangular distribution behavior. The triangular distribution parameters are usually provided by the process owners. A subject matter expert (SME) in the system would know the values of the triangular distributions, namely, the location parameter *a,* the scale parameter (*b-a*), and the shape parameter *m,* where $a<m<b$. The IBS analyst would ask the SME of imprecise values of the three parameters as $[\underline{a},\overline{a}]$, $[\underline{m},\overline{m}]$, and $[\underline{b},\overline{b}]$.

Written in terms of $[\underline{a},\overline{a}]$, $[\underline{m},\overline{m}]$, and $[\underline{b},\overline{b}]$, the three intervals form $2^3 = 8$ possible combinations of the parameters such as those shown in Figure 3-7. The upper and the lower bounds are formed by the lower parameters ($\underline{a}$, $\underline{m}$, and $\underline{b}$) and the upper parameters ($\overline{a}$, $\overline{m}$, and $\overline{b}$.), respectively. All the other six combinations are contained inside the formed boundaries of the bounds.
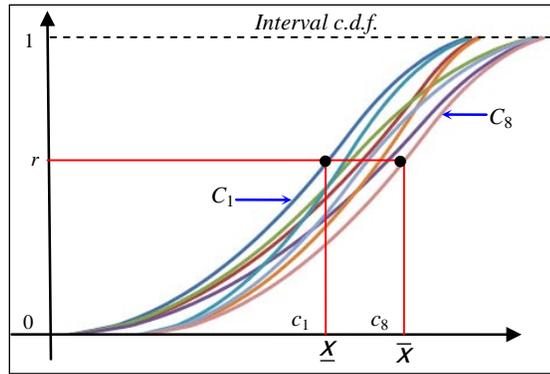
Figure 3-7: Inverse transformation for triangular distributions with interval parameters

3.2.1.2 In the Absence of Data

If no data are available, the analyst can select the distribution type either based on domain experts' opinions, or based on the level of availability or convenience. Note that the distribution does not have to precisely represent the true one. It only bounds the possible ones. The distribution can represent the worst and best scenarios that might occur.

3.2.2 Random Interval Variate Generation

Interval random variates are generated from statistical distributions with interval parameters to run the IBS. Given a statistical distribution with interval parameters, we use the inverse transform to calculate the lower and upper random variables, Here is an example to illustrate the method. Assume a random process that is exponentially distributed as $[\underline{X}, \overline{X}] \sim \exp([\underline{\beta}, \overline{\beta}])$. From the random-number stream $\mathbf{u} = \{u_1, u_2, \dots, u_n\}$, the interval variates are calculated as:

$$[\underline{x_i}, \overline{x_i}] = \left[-\underline{\beta}(\ln(1 - u_i)), -\overline{\beta}(\ln(1 - u_i)) \right] (\forall i \in [1, n])$$

<div align="right">(15)</div>

The obtained interval variates form the lower and the upper *cdf*'s as in Figure 3-2. At a certain value of the cumulative probability $F(x)$, the generated random variate is $[\underline{x}, \overline{x}]$ where $\underline{x}$ and $\overline{x}$ are the lower and upper bounds of the interval random variate, respectively. The upper and lower bounds of *cdf* in Figure 3-2 can also be read in a second way. For a value of a random variable x, the cumulative probability is represented by an interval probability $[\underline{F}(x), \overline{F}(x)]$. These two representations of uncertainty are equivalent.

For distributions with multiple parameters, all combinations of the parameters need to be investigated. Then the respective minimum and maximum from the combinations are selected as the lower and upper bounds of the generated interval random variate. For instance, for a normal distribution with the mean of $[\underline{\mu}, \overline{\mu}]$ and standard deviation of $[\underline{\sigma}, \overline{\sigma}]$. The inverse transform method generates the interval random variate as

$$[\underline{x_i}, \overline{x_i}] = \left[\underline{\mu} + \sqrt{2}\sigma_l \, erf^{-1}(2u_i - 1), \mu + \sqrt{2}\sigma_u \, erf^{-1}(2u_i - 1) \right] (\forall i \in [1, n])$$

<div align="right">(16)</div>

again

$$\sigma_l = \begin{cases} \overline{\sigma}, \text{if } i < n/2 \\ \underline{\sigma}, \text{if } i < n/2 \end{cases} \text{and } \sigma_u = \begin{cases} \underline{\sigma}, \text{if } i < n/2 \\ \overline{\sigma}, \text{if } i < n/2 \end{cases}$$

Similarly, the parameterization of distributions with three or more parameters needs to consider all the possible combinations with respect to the real-valued distributions. For instance, assume a random variable has a triangular distribution

<div align="center">42</div>

triang($[\underline{a},\overline{a}], [\underline{b},\overline{b}], [\underline{m},\overline{m}]$) with interval parameters for the location, the scale, and the shape parameters $[\underline{a},\overline{a}]$, $[\underline{m},\overline{m}]$, and $[\underline{b},\overline{b}]$, respectively. We first assume $[\underline{X},\overline{X}]' \sim$triang($[0,0], [1,1], [\underline{k},\overline{k}]$) where $\underline{k} = (\underline{m} - \underline{a})/(\underline{b} - \underline{a})$ and $\overline{k} = (\overline{m} - \overline{a})/(\overline{b} - \overline{a})$, then the random variate $[\underline{x}_i, \overline{x}_i]$ is calculated as follows,

$$[\underline{x}_i, \overline{x}_i] = \begin{cases} \left[\underline{a} + (\underline{b}-\underline{a})\sqrt{\underline{k}u_i}, \overline{a} + (\overline{b}-\overline{a})\sqrt{\overline{k}u_i}\;\right] & if\ 0 \le u_i \le \underline{k}, 0 \le u_i: \\[2mm] \left[\underline{a} + (\underline{b}-\underline{a})\times\left(1 - \sqrt{(1-\underline{k})(\underline{a}-u_i)}\right), \overline{a} + (\overline{b}-\overline{a})\sqrt{\overline{k}u_i}\;\right] & if\ \underline{k} \le u_i \le 1, 0 \le u_i \\[2mm] \left[\underline{a} + (\underline{b}-\underline{a})\sqrt{\underline{k}u_i}, \overline{a}, \overline{a} + (\overline{b}-\overline{a})\times\left(1 - \sqrt{(1-\overline{k})(\overline{a}-u_i)}\right)\right] & if\ 0 \le u_i \le \underline{k}, \overline{k} \le u_i: \\[2mm] \left[\underline{a} + (\underline{b}-\underline{a})\times\left(1 - \sqrt{(1-\underline{k})(\underline{a}-u_i)}\right), \overline{a} + (\overline{b}-\overline{a})\times\left(1 - \sqrt{(1-\overline{k})(\overline{a}-u_i)}\right)\right] & if\ \underline{k} \le u_i \le 1, \overline{k} \le u_i: \end{cases} \quad (17)$$

Once random intervals are generated, they are used in simulation. Interval arithmetic is applied in calculating interval values. Simulation robustness measure, defined as the required IBS replication length to enclose real-valued *cdf*'s between the IBS bounds with a certain level of confidence, is described in following section.

3.2.3   Simulation Robustness Measure

For the interval enclosure of real-valued *cdf,* a natural question we would like to ask is how much confidence we have that a simulation result based on intervals would include all possible scenarios if real-valued *cdf*'s are used as in the traditional simulation. We need to measure the robustness quantitatively. Figure 3-8 illustrates the simulation enclosure situation.
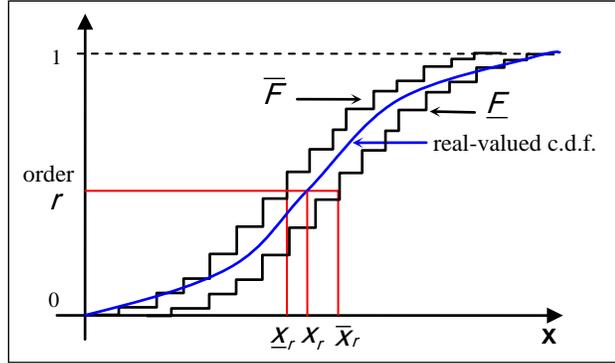
Figure 3-8: An illustration of interval *cdf*

Our objective of reliable simulation is that the generated interval random variates or simulation output performance based on intervals (as empirical *cdf* $[\underline{F}, \overline{F}]$) should include the *unknown cdf* of real-valued distribution. In general, enclosing the small or large orders of observations is more difficult than enclosing those in the middle. Obviously, the probability of the desired enclosure will increase as the replication length increases.

Eq.(5) can be used to measure simulation robustness in terms of the probability of enclosing all possible real-valued scenarios by the interval at a particular order r. Substituting Eq.(4) in Eq.(5) we receive the general form to measure the simulation robustness

$$P\left(\underline{x}_r \leq x_r \leq \overline{x}_r\right) = \sum_{j=r}^{n} \left[\binom{n}{j} \left(\underline{F}(x_r)\right)^j \left(1 - \underline{F}(x_r)\right)^{n-j} \right] \tag{18}$$

$$\times \left(1 - \sum_{j=r}^{n} \left[\binom{n}{j} \left(\overline{F}(x_r)\right)^j \left(1 - \overline{F}(x_r)\right)^{n-j} \right] \right)$$

The robustness measure in Eq.(30) is general and can be applied to both input random variate and output performance enclosure. It can be used to determine the

minimum IBS replication length necessary for a prefixed value of confidence level $(1 - \alpha)$.

Single Parameter Distribution

In Section 3.2 and (Batarseh & Wang, 2008), we derived and implemented the enclosure relationship of Eq.(18) when a stochastic process is exponentially distributed, the result is in

$$P(\underline{x}_r \leq x_r \leq \overline{x}_r) = \sum_{j=r}^{n} \left[\binom{n}{j} \left(1 - \left(1 - \frac{r - 0.5}{n} \right)^{\beta/\underline{\beta}} \right)^{j} \times \left(\left(1 - \frac{r - 0.5}{n} \right)^{\beta/\underline{\beta}} \right)^{n-j} \right] \tag{19}$$
$$\times \left(1 - \sum_{j=r}^{n} \left[\binom{n}{j} \left(1 - \left(1 - \frac{r - 0.5}{n} \right)^{\beta/\overline{\beta}} \right)^{j} \times \left(\left(1 - \frac{r - 0.5}{n} \right)^{\beta/\overline{\beta}} \right)^{n-j} \right] \right)$$

Figure 3-9 illustrates the algorithm to calculate n as the measure of robustness.

<div style="border:1px solid black; padding:10px;">

To calculate the robustness measure n at any order r for an exponential distribution:

Step 0: Given β, $[\underline{\beta}, \overline{\beta}]$, and α

Step 1: Set $n = r$

Step 2: Calculate the probability p in Eq.(19)

Step 3: If $(p \geq 1 - \alpha)$

         return n

      Else $n = n + 1$,

        Back to *Step 2.*

</div>

Figure 3-9: The algorithm to calculate the replication length of the IBS with an exponential distribution

We solve Eq.(19) numerically for three ratios of $\left[\underline{\beta}/\beta, \overline{\beta}/\beta \right]$ as [0.9,1.1], [0.8,1.2] , and [0.6,1.4] and construct Tables A-1, A-2 and A-3 respectively in Appendix

A. The tables can be used based on the ratios between the bounds of the interval mean and the real-valued mean, regardless of the absolute values of the means. The replication lengths were calculated to achieve the confidence levels (CLs) of 90% and 95%. For Table A-1 the replication length was calculated only at 90% CL. The replication lengths needed at 90% CL is very large in general due to the narrow interval of [0.9,1.1]. The replication lengths for 95% CL are even greater than the corresponding ones for 90% CL. It is noticed that as the interval width increases the replication length for simulation decreases at the same percentile, yielding the same probability of enclosure.

In the calculation, we stopped when n is greater than 1030 because the program reaches its computational limit of calculating the large n. In the tables, (> 1030) indicates the limit is reached. The maximum bounding probability when $n = 1030$ is also given in the tables. The transition from > 1030 to three or two decimals of replication numbers shows how affordable it is to reach the completeness of the solution in these orders of r. For the small orders of observations, the replication required for a specified CL is very large. It shows the difficulty of enclosing the real-valued *cdf* at small orders of r for small interval widths compared to large interval widths. For the very large orders of observations, the replication length also starts to increase. This is due to the narrow width of the *cdf* bounds at the high cumulative probability as the *cdf* curves become flatter.

Multiple Parameter Distributions

The simulation robustness measure is studied for the normal and the triangular distributions as multiple parameter distributions.

Normal Distribution

The simulation robustness measure can be also applied to the multiple parameter distributions. For instance, Eq.(18) can be rewritten for the normal distribution with a mean of $[\underline{\mu}, \overline{\mu}]$ and a standard deviation $[\underline{\sigma}, \overline{\sigma}]$, as follows:

$$P(\underline{x_r} \le x_r \le \overline{x}_r) = \left(\sum_{j=r}^{n} \left[\binom{n}{j} \left(\frac{1}{2} + \frac{1}{2} erf \left(\frac{x_r - \underline{\mu}}{\sigma_l/\sqrt{2}} \right) \right)^j \times \left(\frac{1}{2} - \frac{1}{2} erf \left(\frac{x_r - \underline{\mu}}{\sigma_l/\sqrt{2}} \right) \right)^{n-j} \right] \right) \qquad (20)$$

$$\times \left[1 - \left(\sum_{j=r}^{n} \left[\binom{n}{j} \left(\frac{1}{2} + \frac{1}{2} erf \left(\frac{x_r - \overline{\mu}}{\sigma_u/\sqrt{2}} \right) \right)^j \times \left(\frac{1}{2} - \frac{1}{2} erf \left(\frac{x_r - \overline{\mu}}{\sigma_u/\sqrt{2}} \right) \right)^{n-j} \right] \right) \right]$$

The values of σ_l and σ_u are set based on the corresponding order r as discussed above and x_r are calculated as in Eq.(16). Yet again, the simulation length required for the normal distribution is calculated from Eq.(33) as the value n that guarantees the enclosure of the real variate at any order r with a certain level of confidence. Given the values of $[\underline{\mu}, \overline{\mu}]$ and $[\underline{\sigma}, \overline{\sigma}]$, we use Eq.(20) to find n by gradually increasing its value until its satisfies the desired probability of enclosure of $(1 - \alpha)$ at any order r as shown in Figure 3-10.

To calculate the robustness measure n at any order r
for the normal distribution:

Step 0: Given μ, σ , $[\underline{\mu}, \overline{\mu}]$, $[\underline{\sigma}, \overline{\sigma}]$, and α

Step 1: Set $n = r$

Step 2: Calculate the probability p in Eq.(20)

Step 3: If $(p \geq 1 - \alpha)$

　　　　return n

　　Else $n = n + 1$,

　　　　Back to *Step 2*.

Figure 3-10: The algorithm to calculate the replication length of the IBS for the normal distribution

Triangular Distribution

In addition, the probability in Eq.(16) can be used for the triangular distribution function with the three parameters as $[\underline{a}, \overline{a}]$, $[\underline{m}, \overline{m}]$, and $[\underline{b}, \overline{b}]$. The upper triangular *cdf* using the lower parameters replaces $\underline{F}(x)$ and the lower *cdf* replaces $\overline{F}(x)$ using the upper parameters as follows:

1.　　If $a \leq x_r \leq m$ then Eq.(18) is derived as follows:

$$P\left(\underline{x}_r \leq x_r \leq \overline{x}_r\right) = \sum_{j=r}^{n} \left[\begin{array}{c} \binom{n}{j}\left(\left(a + (b-a)\sqrt{k((r-0.5)/n)} - \underline{a}\right)^2 / [(\underline{b}-\underline{a})(\underline{m}-\underline{a})]\right)^j \\ \left(1 - \left(a + (b-a)\sqrt{k((r-0.5)/n)} - \underline{a}\right)^2 / [(\underline{b}-\underline{a})(\underline{m}-\underline{a})]\right)^{n-j} \end{array} \right] \tag{21}$$

$$\times \left[1 - \sum_{j=r}^{n} \left[\begin{array}{c} \binom{n}{j}\left(\left(a + (b-a)\sqrt{k((r-0.5)/n)} - \overline{a}\right)^2 / [(\overline{b}-\overline{a})(\overline{m}-\overline{a})]\right)^j \\ \left(1 - \left(a + (b-a)\sqrt{k((r-0.5)/n)} - \overline{a}\right)^2 / [(\overline{b}-\overline{a})(\overline{m}-\overline{a})]\right)^{n-j} \end{array} \right] \right]$$

2.   If $m < x_r \leq b$ then Eq.(18) is derived as follows:

$$P(\underline{x}_r \leq x_r \leq \overline{x}_r) = \sum_{j=r}^{n} \left[\frac{\binom{n}{j}\left(1 - \left(\underline{b} - a + (b-a)\left[1 - \sqrt{(1-k)(1-(r-0.5)/n)}\right]\right)^2 / \left[(\underline{b} - \underline{a})\left(\underline{b} - \underline{m}\right)\right]\right)^{j}}{\left(1 - \left(\underline{b} - a + (b-a)\left[1 - \sqrt{(1-k)(1-(r-0.5)/n)}\right]\right)^2 / \left[(\underline{b} - \underline{a})\left(\underline{b} - \underline{m}\right)\right]\right)^{n-j}} \right]$$
$$\times \; 1 - \sum_{j=r}^{n} \left[\frac{\binom{n}{j}\left(1 - \left(\overline{b} - a + (b-a)\left[1 - \sqrt{(1-k)(1-(r-0.5)/n)}\right]\right)^2 / [(\overline{b} - \overline{a})(\overline{m} - \overline{a})]\right)^{j}}{\left(1 - \left(\overline{b} - a + (b-a)\left[1 - \sqrt{(1-k)(1-(r-0.5)/n)}\right]\right)^2 / [(\overline{b} - \overline{a})(\overline{m} - \overline{a})]\right)^{n-j}} \right] \qquad (22)$$

Eq.(21) and Eq.(22) are used according to real-valued random variate value x_r

w.r.t. real-point parameters, where x_r is calculated as in Eq.(17). The required replication

length n is also calculated as discussed previously and as shown in Figure 3-11.

<div style="border:1px solid black; padding:10px;">

To calculate the robustness measure n at any order rfor a triangular distribution:

Step 0: Given $a, m, b, [\underline{a}, \overline{a}]$ $[\underline{m}, \overline{m}]$, and $[\underline{b}, \overline{b}]$, and α

Step 1: Set $n = r$

Step 2: Calculate the probability p in Eq.(21) or (22)

Step 3: If $(p \geq 1 - \alpha)$

     return n

   Else $n = n + 1$,

    Back to *Step 2.*

</div>

Figure 3-11: The algorithm to calculate the replication length of the IBS for the triangular distribution

3.3   Uncertainty Propagation in the IBS

In a typical DES model, an entity (i) is created by a Source at time a_i based on statistical distributions. Then the entity starts its life cycle in a system and ends at a Sink. Statistics are collected, such as how much time the entity spent in the system. In contrast, in the IBS, an entity (i) arrives at the system with an interval arrival time $[\underline{a}_i, \overline{a}_i]$. This interval represents the uncertainty associated with the arrival time. Each entity then is assigned to spend an interval service time $[\underline{s}_i, \overline{s}_i]$ at each station in the system. For instance, a simple linear system with interval random variables, where interested random variables such as arrival and service times are random intervals, is shown in Figure 3-12.

| Source Arrival time: Expo($[\underline{\lambda},\overline{\lambda}]$) | Station 1 Service time: N($[\underline{\mu},\overline{\mu}]$,σ) | Station 2 Service time: N($[\underline{\mu},\overline{\mu}]$,$[\underline{\sigma},\overline{\sigma}]$) | Station K | Sink |
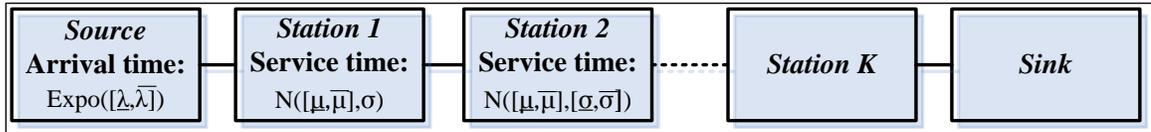
Figure 3-12: Simple Linear System Based on Intervals

In the traditional simulation, entity (i) arrives at a linear system with K stations such as the one in Figure 3-12 at time a_i and departs at time d_i after spending a total time of $s_i = \sum_{k=1}^{K} s_{ki}$, where s_{ki} is the time entity (i) spends at station k. The performance measures are calculated based on real arithmetic that assumes no uncertainty is included in the entity's time attributes. However, the IBS is based on interval arithmetic that accounts for the uncertainty propagation. For instance, the total time in the system here is calculated as $[\underline{s}_i, \overline{s}_i] = [\sum_{k=1}^{K} \underline{s}_{ki}, \sum_{k=1}^{K} \overline{s}_{ki}]$ in worst-case. The width of an interval represents the level of uncertainty. In the worst-case scenario, the uncertainty associated with the total time is greater than those of times in individual stations. The performance

measures are also calculated based on interval arithmetic that estimates the best and worst-case scenarios. The interval estimations assist in evaluating the completeness and the soundness of our solution. With random intervals generated, the simulation starts with a proposed simulation clock advancement mechanism, which is addressed in Chapter 4 with the [M]/[M]/1 example.

CHAPTER 4:
DEMONSTRATION OF SIMULATION CLOCK MECHANISM AND THE IBS IMPLENATION IN JSIM

This chapter discusses handling the simulation events to advance the simulation clock in the IBS. In particular, we need to investigate possible approaches that could be employed to handle the simulation events in the IBS. Thus, this chapter first introduces the nature of the problem related to the simulation clock advancement in the IBS as discussed in Section 4.1. Section 4.2 proposes three possible approaches to sort the simulation events and accordingly advance the clock based on lower times, upper times, and a uniform sampling. Section 4.3 proposes new statistical dispersion measures for interval data. The implementation of the IBS in *JSim* is discussed in Section 4.4. Finally, a hand simulation of [M]/[M]/1 is illustrated in Section 4.5 to demonstrate the IBS mechanism.

4.1   Simulation Clock advancement in the IBS

The simulation clock is the simulation model variable that gives the current value of the simulated time. Based on the next-event-time advancement approach, the simulation clock is advanced from one event time to the next scheduled one. One of the greatest new challenges in the IBS is the advancement of the simulation clock. In the IBS, the events are scheduled to happen within a window time represented by an interval $[\underline{t_i}, \overline{t_i}]$. In the traditional DES, it is easy to decide the next event that will advance the simulation clock. However, the partial order between two intervals in the IBS is more complex than real numbers. In other words, the less-than-or-equal-to relationship

between two arrival times $[\underline{a}_i, \overline{a}_i]$ and $[\underline{a}_{i+1}, \overline{a}_{i+1}]$ for entities i and $i + 1$ can be defined in several ways. There are six cases in which two intervals, $[\underline{a}_i, \overline{a}_i]$ and $[\underline{a}_{i+1}, \overline{a}_{i+1}]$, are located w.r.t. each other, as illustrated in Figure 4-1. In simulation, events are stacked in a so-called *event list* in an ascending order based on their time of occurrence. When events with interval times are inserted in the event list, it is difficult to prioritize them if they occurred as in the cases of 3, 4, 5, and 6 because the two intervals overlap. That is, given existing uncertainties in the system, there may not be a single, clear, or best approach to handle the events in the IBS. Therefore, we counter this by analyzing three possible approaches to handle the events.
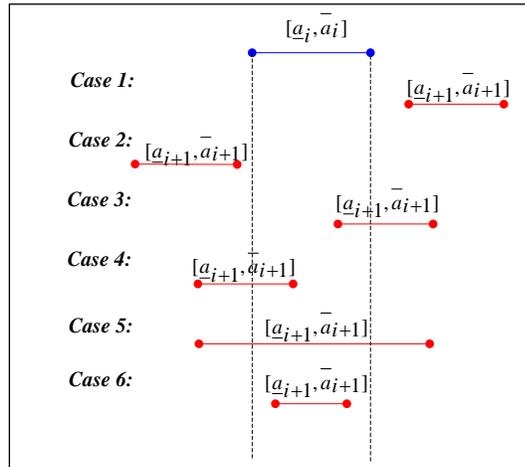
Figure 4-1: Six locations of two intervals with respect to each other

In Figure 4-1, the event list consists of interval events. Therefore, we define the simulation clock variable in the IBS as an *interval simulation clock* initiated at time $[0,0]$ and the interval times of future events determine the clock time. The lower and upper bounds of the simulation clock tracks the earliest possible and the latest possible times of

53

the events, respectively. For instance, when it is time to execute the next event, the event at the top of the event list is removed based on some event selection rules, and the interval simulation clock is advanced, thereby synchronizing the lower and the upper bounds.

There are multiple approaches that could be employed to sort simulation events in the IBS platform. We propose three possible approaches to handle events. One approach is to use the lower interval times. In other words, we prioritize the events in the event list based on their earliest possible occurrence time. We refer to this approach as *lower-based sorting*. Another possible approach is to prioritize the events based on their upper bounds. This approach sorts the events based on their latest possible time of occurrence and is referred to as *upper-based sorting*. Finally, a third approach is to handle the events based on a uniformly sampled time from the events' interval times. The details of the three clock advancement approaches will be described in Section 4.2. Herein, we illustrate how IBS runs by the [M]/[M]/1 example. Regardless of the selected approach, there are three events involved in the simulation of the [M]/[M]/1 example. They are:

1) Arrival: entity (i) enters the system at time $[\underline{a}_i, \overline{a}_i]$;

2) Departure: entity (i) leaves the system at time $[\underline{d}_i, \overline{d}_i]$ after its service is completed;

3) End: the simulation stops after a designated time.

For all of the three proposed clock advancement approaches, the simulation event list is structured based on a real-valued arrival time obtained from the interval arrival

times $[\underline{a}_i, \overline{a}_i]$'s. The following is the next-event simulation algorithm that is executed in the IBS:

- *Initialize*: interval simulation clock is initialized at [0,0].

- *Process Event and Advance Clock*: the most imminent event is processed and the simulation clock is updated according to the occurrence time of the scheduled event $[\underline{t}_{now}, \overline{t}_{now}]$. This update accounts for the uncertainty associated with the events occurrence times.

- *Schedule Next Event*: a new event is selected from the event list based on a pre-determined approach to replace the processed event, and the algorithm goes to Step (2) to process new events.

- *Terminate Simulation*: simulation continues to process next events until a terminal condition is satisfied.

This is the general algorithm behind the IBS when used for queueing systems simulations based on the next-event-time advancement approach. A natural question arises at this point: does the employment of one single real-valued instance from the time intervals neglect the possibility of entity $(i + 1)$ arriving earlier than entity (i) with an overlap of their interval times? We answer this question by analyzing the result of handling the event list based on the three clock advancement approaches as in section 4.2.

4.2 Proposed Approaches to Advance the Simulation Clock in IBS

As discussed earlier, three possible approaches to handle the event list in the IBS are investigated. First, let us assume the interval arrival times of entities i and $i + 1$ are given as $[\underline{a}_i, \overline{a}_i]$ and $[\underline{a}_{i+1}, \overline{a}_{i+1}]$, respectively, and they are given as proper intervals. After the entities get sorted in the event list, their service-start time becomes the critical factor to be determined. The service-start time of entity (i), $[\underline{sst}_i, \overline{sst}_i]$, is calculated as

$$[\underline{sst}_i, \overline{sst}_i] = [\max(\underline{a}_i, \underline{d}_{i-1}), \max(\overline{a}_i, \overline{d}_{i-1})] \tag{23}$$

Eq.(23) is based on the maximum of entity (i) arrival time and entity $(i - 1)$ departure time for each bound separately. Eq.(23) represents the initial time interval attribute that is attached to the entities upon their arrivals. The entity's service-start time changes in accordance with the selected approach to sort the event list. As a result, the obtained interval performance measures vary. Herein, the three proposed approaches are discussed and their resulting effects on the performance measures of interest are analyzed.

4.2.1 Lower bound Approach to Advance Simulation Clock

The first approach is the lower-based sorting. This approach manages the event list based on the earliest events time of occurrence. In this respect, the simulation events are prioritized based on the lower bounds of times. Consequently, the event list is comprised of any two events stacked as in cases of 1, 3, and 6 of Figure 4-1.

In case 1, interval $[\underline{a}_i, \overline{a}_i]$ is associated with interval $[\underline{a}_{i+1}, \overline{a}_{i+1}]$ in a *less than* relationship "$<$" defined as

$$[\underline{a}_i, \overline{a}_i] < [\underline{a}_{i+1}, \overline{a}_{i+1}] \Leftrightarrow (\underline{a}_i < \underline{a}_{i+1}) \wedge (\overline{a}_i < \underline{a}_{i+1}) \wedge (\overline{a}_i < \overline{a}_{i+1}) \tag{24}$$

For case 3, interval $[\underline{a}_i, \overline{a}_i]$ is associated with interval $[\underline{a}_{i+1}, \overline{a}_{i+1}]$ in a *partially less than* relationship "$\prec$" defined as

$$[\underline{a}_i, \overline{a}_i] \prec [\underline{a}_{i+1}, \overline{a}_{i+1}] \Leftrightarrow (\underline{a}_i < \underline{a}_{i+1}) \wedge (\overline{a}_i > \underline{a}_{i+1}) \wedge (\overline{a}_i < \overline{a}_{i+1}) \tag{25}$$

In both cases the condition $(\underline{a}_i < \underline{a}_{i+1}) \wedge (\overline{a}_i < \overline{a}_{i+1})$ is satisfied. In other words, the lower and upper arrival times are arranged in an ascending order. Consequently, there is no disorder in events sorting at either bound. However, case 6 leads to a logical complexity because interval $[\underline{a}_i, \overline{a}_i]$ is associated with interval $[\underline{a}_{i+1}, \overline{a}_{i+1}]$ in an *inclusion* relationship "$\supset$" defined as

$$[\underline{a}_i, \overline{a}_i] \supset [\underline{a}_{i+1}, \overline{a}_{i+1}] \Leftrightarrow (\underline{a}_i < \underline{a}_{i+1}) \wedge (\overline{a}_i > \overline{a}_{i+1}) \tag{26}$$

In this case, the entity placed ahead in the event list has a larger upper arrival than of the succeeding one, i.e. $(\overline{a}_i > \overline{a}_{i+1})$. Therefore, the entities at their upper arrival times are not served according to the *first-in-first-out* (FIFO) basis as desired. This disorder in the upper times commonly causes an increase in the minimum expected performance measures as addressed below.

In queueing systems, the lower-based sorting can be compared to a traditional simulation experiment with the higher arrival rate as the input. A simulation run with the lower arrival rate estimates a *worst-case scenario* w.r.t. the waiting time in the queue (i.e. waiting time in the event list) referred to as w_u. To illustrate, consider the bank example in Figure 3-1 with interval- arrival rate of $[\underline{\lambda}, \overline{\lambda}]$ and a service rate of $[\underline{\mu}, \overline{\mu}]$. Assume a

simulation analyst runs a traditional simulation using the lower parameters of $\underline{\lambda}$ and μ. In this scenario, the arrival rate $\underline{\lambda}$ is the mean and the standard deviation of the arrival rate that is exponentially distributed. This higher arrival rate $\bar{\lambda}$ generates smaller and less dispersed inter-arrival times when compared to the lower rate $\underline{\lambda}$. Therefore, in the traditional sense, a worst-case scenario with respect to the average waiting time in queue is estimated.

On the other hand, running a traditional simulation experiment using the upper parameters of the bank example, $\underline{\lambda}$ and μ, estimates the best-case scenario w.r.t. the average waiting time in queue, given as w_l. In this scenario, the event list is created based on inter-arrival times that are generated from an exponential distribution with a smaller rate $\underline{\lambda}$. Therefore, the resulted service start time represents the *best-case scenario with respect to the average performance measures* in the traditional simulation. The lack of capturing the input uncertainties in the traditional simulation assumes that the only use of the lower arrival rate estimates a best-case scenario w.r.t. performance measures in the context of queueing systems. Analogously, the use of the upper arrival rate estimates the *worst-case scenario*. Figure 4-2 demonstrates the best- and worst- case scenarios in the traditional simulation.
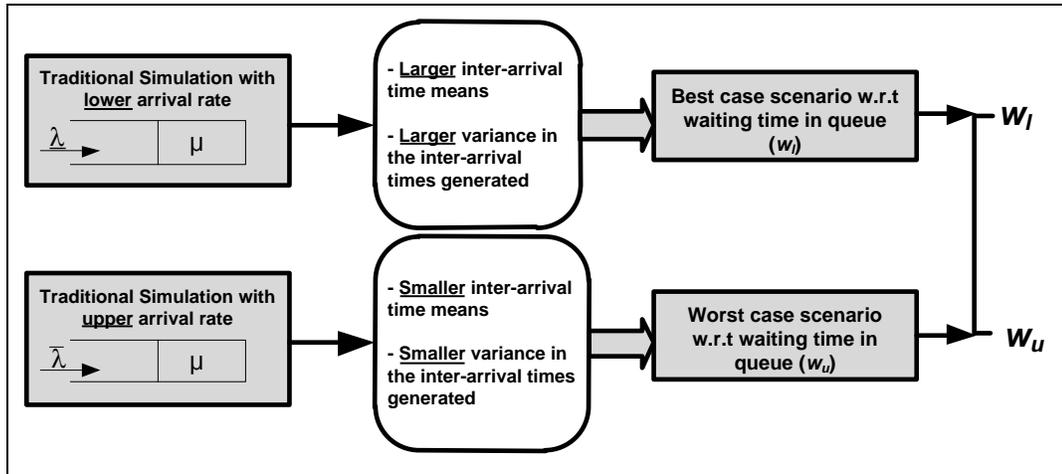
Figure 4-2: Best- and worst-case scenarios in the traditional sense

In the view of the IBS, the interval parameters assume that any parameter value enclosed within the interval parameter (i.e. $\lambda \in [\underline{\lambda}, \overline{\lambda}]$) is valid and credible to estimate an expected scenario, even the lower and the upper parameters. The IBS analyst does not read the best and worst-case scenarios in the traditional manner. The best and worst-case scenarios in the IBS are judged based on the yielded uncertainties in the estimated results. In other words, if the uncertainty of performance measures given as $\text{wid}(w) = |\overline{w} - \underline{w}|$ is greater, this reports a worst-case scenario. On the contrary, as the width measure decreases, this estimates a best-case scenario.

In the IBS lower-based sorting, if there is no disorder encountered in the lower arrival times, the upper bound of performance measures estimate will be the same as w_u in the traditional simulation, i.e. $\overline{w} = w_u$. However, when there is a disorder in the upper arrival times caused by case 6 in Figure 4-1, this disorder tends to increase the performance measure of interest at the lower bound $\underline{w}$, i.e. $\underline{w} > w_l$. The increase in the lower bound estimate is due to the disorder caused by case 6. The disorder is a source of

59

variability that tends to increase the estimated performance measure than its expected value, w_l. The increase in the lower performance measure reduces the width of the interval performance measures and this estimate is assessed as the best-case scenario of the IBS because the uncertainty reduces.

However, when the analyst face an imprecise service rate given as $[\underline{\mu}, \overline{\mu}]$, a dual operator could be used for adding the service time to the lower service-start time in order to estimate the best-case scenario w.r.t. the other performance measures of interest. For instance, a service time, given as a proper interval $[\underline{s}_i, \overline{s}_i]$, represents an imprecise service time of entity (i). If the IBS follows the lower-based sorting and the service-start time of entity (i) is given as $[\underline{sst}_i, \overline{sst}_i]$, then the worst-case and best-case sojourn time can be as in Eq.(27) and Eq.(28),

$$[\underline{t}_i, \overline{t}_i] = [\underline{d}_i, \overline{d}_i] - [\underline{a}_i, \overline{a}_i] \tag{27}$$

$$[\underline{t}_i, \overline{t}_i] = [\underline{d}_i, \overline{d}_i] - dual([\underline{a}_i, \overline{a}_i]) \tag{28}$$

respectively. With the interval addition operation as shown in Eq.(27), the lower bound of sojourn time is derived by subtracting the upper bound of arrival time from the lower bound of departure time. The upper bound is calculated from the opposite bounds. Therefore, Eq.(27) estimates an imprecise worst-case scenario with respect to sojourn time, whereas Eq.(28) gives the best-case estimations.

In summary, if the simulation analyst is interested in a best-case scenario, lower-based sorting should be adopted. Moreover, a *dual* operator can be used as appropriate to estimate an imprecise best-case scenario w.r.t. the performance measures of interest.

4.2.2   Upper Bound Approach to Advance Simulation Clock

A second possible approach is to handle the simulation based on the events latest possible times. If the upper-based sorting is followed, the IBS event list is composed of the cases 1, 3, and 5 for any two subsequent events as in Figure 4-1. When compared to the lower-based sorting, it seems that the only difference in the event list is cases 5 and 6 for the lower and the upper bounds, respectively. However, the obtained sequences in the event list based on either bound are not the same. For instance, the first event placed in the event list might not be the same if the lower-based sorting was used over the upper-based or vice-versa. Thus, all next scheduled events generate two different sequences for any of the two bounds based sorting approaches.

Cases 1 and 3 occur in a similar aspect within the lower-based and the upper-based sorting. The relationships between two events occur as in cases 1 and 3 are discussed in Section 4.2.1. Clearly, there is no disorder arises at any bound, i.e. ($\underline{a}_i < \underline{a}_{i+1}) \land (\overline{a}_i < \overline{a}_{i+1})$. However, in case 5, interval $[\underline{a}_i, \overline{a}_i]$ is associated with interval $[\underline{a}_{i+1}, \overline{a}_{i+1}]$ in an *inclusion* relationship "$\subset$" defined as

$$[\underline{a}_i, \overline{a}_i] \subset [\underline{a}_{i+1}, \overline{a}_{i+1}] \Leftrightarrow (\underline{a}_i > \underline{a}_{i+1}) \land (\overline{a}_i < \overline{a}_{i+1}) \tag{29}$$

In this case, entity (i) that is placed ahead in the event list has a lower arrival time that is larger than of the succeeding entity ($i + 1$), i.e. $(\underline{a}_i > \underline{a}_{i+1})$. Therefore, the entities are not served according to the FIFO basis at their lower bounds. The disorder arises from case 5 as the lower bound increases the expected value of upper bound of waiting time $\overline{w}$ from the traditional simulation with upper parameter values (arrival rate $=\overline{\lambda}$ and service rate $=\mu$), i.e. $\overline{w} > w_u$. Under this, there are no disorders at the upper

bound and the resulted lower waiting time is equal to the yielded waiting time from the traditional simulation with the lower parameters (arrival rate $=\underline{\lambda}$ and service rate $=\mu$), i.e. $\underline{w} > w_l$. That is, given the upper-based sorting approach, the imprecise estimate of the IBS have an interval width that is greater than the difference between the traditional simulation estimates with lower and upper parameters, i.e. $wid(w) = |\overline{w} - \underline{w}| > (w_u - w_l)$. This greater uncertainty in the estimates provides a worst-case scenario with respect to waiting time in system. With this in mind, the *dual* operator could be used in interval arithmetic for the performance measures to estimate their imprecise worst-case scenario.

The difference between the lower-based and the upper-based sorting are organized in Figure 4-3 as a nested hierarchy. In this graphical illustration, we refer to the traditional simulation estimates by w_l and w_u as illustrated in Figure 4-2. Because of this lack of modeling uncertainties in traditional simulation, these estimates are levied as the minimum and maximum waiting time in queue. By performing the traditional simulation, the analyst is in essence checking the extreme values of the waiting time in queue as indicated using the green horizontal bar in Figure 4-3.

Under this, the lower-based sorting in the IBS of queueing systems estimates an upper waiting time as $\overline{w} = w_u$. Yet, the modeled uncertainties causes a disorder at the lower bound that increases its waiting time from the minimum waiting time estimated from the traditional simulation, i.e. $\underline{w} > w_l$. This disorder is encountered due to the occurrence of case 6 (refer to Figure 4-1) in the event-list. Moreover, the lower-based sorting produces a more conservative estimate than the traditional extremes values and is considered a best-case scenario of the IBS. The best-case scenario is displayed in Figure 4-3 using a horizontal blue bar that is range is tighter than the green one.

62

On the other hand, the IBS upper-based sorting confounds the upper waiting time from the maximum waiting time obtained from the traditional simulation, i.e. $\overline{w} > w_u$. More formally, the interval bounds here $[\underline{w}, \overline{w}]$ are wider than the traditional simulation range of estimates, i.e. $(\overline{w} - \underline{w}) > (w_u - w_l)$. This disorder is resulted because of the occurrence of case 5 (refer to Figure 4-1) in the event list. The over estimation of the interval performance measures characterizes the upper-based sorting as an approach to estimate the worst-case scenario of queuing systems. The red horizontal bar represents the interval estimation of the worst-case scenario whose range is wider than the green bar. Finally, we mention the use of dual operator to estimate the best- and worst-case scenarios w.r.t. the other performance measures than the waiting time in queue.
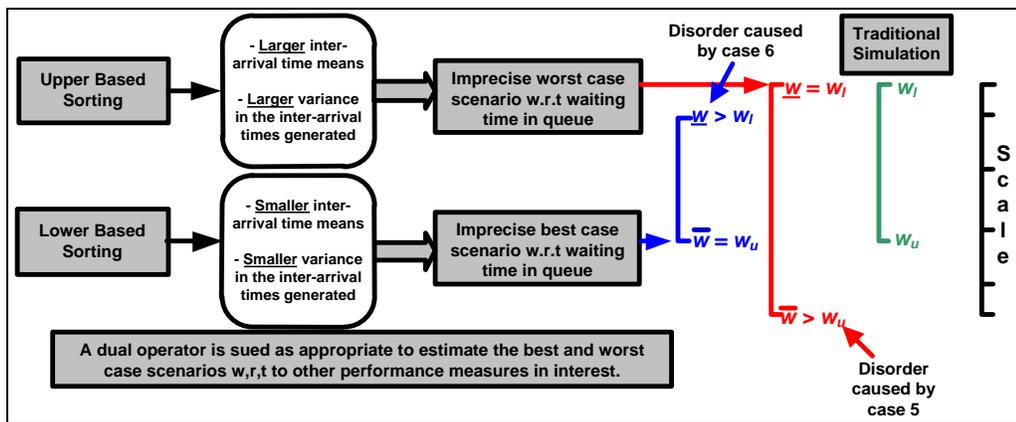
Figure 4-3: Lower-based vs. Upper-based Event List Sorting in the IBS

This shows that the explicit and simultaneous consideration of uncertainty and variability in simulation yields reliable simulation results in one run which can improve the decision making process. More specific to the context of queueing systems, we

differentiate between the two sorting approaches as follows. An upper-based sorting is preferred over a lower-based sorting if the worst-case scenario is of interest. On the other hand, the lower-based sorting is favorable to estimate the best-case scenario w.r.t imprecise performance measures. This difference is revealed because smaller arrival rate with smaller variability estimates higher values of the performance measures, on the other hand, larger arrival rate with larger variability results into smaller values of the performance measures in the queueing systems. Moreover, the above illustration demonstrates how this fact imposes the modeling of the best- and worst-case scenarios in the IBS.

4.2.3   Time Sampled Approach to Advance Simulation Clock

A third proposed approach to advance the clock is based on sampled instances from the interval times. With the assumption that the interval times are uniformly distributed, the entities are sorted based on its sample time. Consequently, each simulation event is attributed with two time formats, an interval time $[\underline{t}_i, \overline{t}_i]$ and a real-valued time $t_i \sim U(\underline{t}_i, \overline{t}_i)$. These sampled values $t_i's$ are used to sort the event in the event list. As a result, the events may occur in different sequences because of the extra uniform sampling procedure adopted to prioritize the events.

Based on the sampling time approach, the probability of advancing interval $[\underline{a}_i, \overline{a}_i]$ before interval $[\underline{a}_{i+1}, \overline{a}_{i+1}]$ is studied for the six cases in Figure 4-1. The three factors that have an influence on advancing interval $[\underline{a}_i, \overline{a}_i]$ ahead of $[\underline{a}_{i+1}, \overline{a}_{i+1}]$ in simulation are:

1.   The overlapping case between the two intervals,

2.   The uncertainty associated within the intervals, i.e. intervals' widths. For example, the uncertainty associated within the interval of $[\underline{a}_i, \overline{a}_i]$ is quantified as $wid(a_i) = |\overline{a}_i - \underline{a}_i|$.

3.   The intersection period between the intervals if overlapped.

The probability of advancing interval $[\underline{a}_i, \overline{a}_i]$ ahead of $[\underline{a}_{i+1}, \overline{a}_{i+1}]$ for all cases in Figure 4-1, i.e. probability of FIFO, is studied below.

A ratio distribution (Golberg, 1984) is applied here to find the desired probability of FIFO as explained above. This distribution is essentially constructed from the ratios of two uniformly random variables. Primarily, we examine the probability of advancing event X ahead of event Y, whereas the density functions of the events occurrence times are $f_{T_x}(t_x)$ and $f_{T_y}(t_y)$, respectively. Suppose, $f_{T_x}(t_x)$ and $f_{T_y}(t_y)$ are two continuous uniform distribution functions. In addition, the distribution functions have the same parameters, where the minimum value is l and the maximum is k and moreover, $l, k \geq 0$. Therefore, the occurrence time of the events are sampled as $t_x \sim U(l, k)$ and $t_y \sim U(l, k)$. Then, the ratio of the random variables is $U = T_x/T_y$ which *pdf* is

$$f_u(u) = \begin{cases} \dfrac{1}{2(k-l)^2}\left(k^2 - \dfrac{l^2}{u^2}\right), & \dfrac{l}{k} \leq u < 1 \\ \dfrac{1}{2(k-l)^2}\left(\dfrac{k^2}{u^2} - l^2\right), & 1 \leq u \leq \dfrac{k}{l} \end{cases} \tag{30}$$

The derivation of Eq.(30) is explained as follows. First, the cumulative function of the variable U given by $F_U(u) = P\{U \leq u\}$ can be expressed as:

$$P\{U \leq u\} = \int_G f_{T_x}(t_x) f_{T_y}(t_y) dt_x dt_y$$

65

where $G = \{(t_y, t_x) : t_x/t_y \le u\}$. Then $G = G_1 \cup G_2$, where $G_1 = \{(t_y, t_x) : t_x \le ut_y, t_x \le 1/(k-l)\}$ and $G_2 = \{(t_y, t_x) : t_x \le ut_y, t_x > 1/(k-l)\}$. Thus,

$$P\{U \le u\} = \int_{G_1} f_{T_x}(t_x)\, f_{T_y}(t_y)\, dt_x dt_y + \int_{G_2} f_{T_x}(t_x)\, f_{T_y}(t_y)\, dt_x dt_y \tag{31}$$

We evaluate Eq.(31) for $(l/k \le u \le 1)$ and for $(1 \le u \le k/l)$. The regions G_1 and G_2 are shown in Figure 4-4.
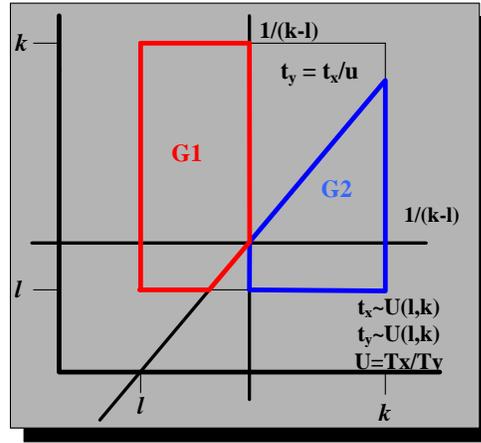
Figure 4-4: Integration region for calculating the distribution of T_x/T_y

First, the evaluation is over $(\frac{l}{k} \le u < 1)$. The double integration gives

$$F_u(u) = \int_{l}^{ku} \int_{\frac{1}{u}t_x}^{k} \frac{1}{(k-l)^2} dt_y\, dt_x = \frac{1}{(k-l)^2} \int_{l}^{ku} \left(k - \frac{1}{u}t_x\right) dt_x = \frac{1}{(k-l)^2}\left[\frac{k^2 u}{2} - lk + \frac{l^2}{2u}\right] \tag{32}$$

and the derivation of the cumulative density function with respect to u gives the probability density function as

$$f_u(u) = \frac{\partial F}{\partial u} = \left(\frac{1}{(k-l)^2}\left[\frac{k^2 u}{2} - lk + \frac{l^2}{2u}\right]\right)' = \frac{1}{(k-l)^2}\left[\frac{k^2}{2} - \frac{l^2}{2u^2}\right] \tag{33}$$

Second, the evaluation is over $(1 \le u \le k/l)$, which results in

$$F_u(u) = 1 - \iint_{lu\,l}^{k\,\frac{1}{u}t_x} \frac{1}{(k-l)^2}\,dt_y\,dt_x = 1 - \frac{1}{(k-l)^2}\int_{lu}^{k}\left(\frac{1}{u}t_x - 1\right)dx = 1 - \frac{1}{(k-l)^2}\left[\frac{k^2}{2u} - lk + \frac{l^2u}{2}\right] \tag{34}$$

Similarly, the derivation of the above cumulative density function gives

$$f_u(u) = \frac{\partial F}{\partial u} = \left(\frac{1}{(k-l)^2}\left[\frac{k^2}{2u} - lk + \frac{l^2u}{2}\right]\right) = \frac{1}{(k-l)^2}\left[\frac{k^2}{2u^2} - \frac{l^2}{2}\right]' \tag{35}$$

Hence, the probability distribution function of u can be summarized as

$$f_u(u) = \begin{cases} \dfrac{1}{2(k-l)^2}\left(k^2 - \dfrac{l^2}{u^2}\right), & \dfrac{l}{k} \le u < 1 \\[3mm] \dfrac{1}{2(k-l)^2}\left(\dfrac{k^2}{u^2} - l^2\right), & 1 \le u \le \dfrac{k}{l} \end{cases} \tag{36}$$

From Eq.(36), we notice that advancing X ahead of Y has equal probability to advancing Y prior to X. This observation is because the ratio of their occurrence instants is equal to one-half. i.e. $P(u < 1) = 0.5$. Eq.(36) is used to calculate the probability of advancing entity (i) with an arrival time $[\underline{a}_i, \overline{a}_i]$ ahead of entity ($i + 1$) with an arrival time as $[\underline{a}_{i+1}, \overline{a}_{i+1}]$ for all the cases as in Table 4-1, where $a_i \sim U(\underline{a}_i, \overline{a}_i)$ and $a_{i+1} \sim U(\underline{a}_{i+1}, \overline{a}_{i+1})$.

TABLE 4-1
THE PROBABILITY OF ADVANCING INTERVAL $[\underline{a}_i, \overline{a}_i]$ PRIOR TO INTERVAL $[\underline{a}_{i+1}, \overline{a}_{i+1}]$ FOR THE SIX CASES IN FIGURE 4-1

Cases	$P(a_i/a_{i+1} < 1)$
1	1
2	0
3	$1 - 1/2 \times (\overline{a}_i - \underline{a}_{i+1})/(\overline{a}_i - \underline{a}_i) \times (\overline{a}_i - \underline{a}_{i+1})/(\overline{a}_{i+1} - \underline{a}_{i+1})$
4	$1/2 \times (\overline{a}_{i+1} - \underline{a}_i)/(\overline{a}_i - \underline{a}_i) \times (\overline{a}_{i+1} - \underline{a}_i)/(\overline{a}_{i+1} - \underline{a}_{i+1})$
5	$1/2 \times (\overline{a}_i - \underline{a}_i)/(\overline{a}_{i+1} - \underline{a}_{i+1}) + (\overline{a}_{i+1} - \overline{a}_i)/(\overline{a}_{i+1} - \underline{a}_{i+1})$
6	$1/2 \times (\overline{a}_{i+1} - \underline{a}_{i+1})/(\overline{a}_i - \underline{a}_i) + (\overline{a}_{i+1} - \overline{a}_i)/(\overline{a}_i - \underline{a}_i)$

Table 4-1 demonstrates that this sampled time approach may result in various sequences of the entities in the event list. When an overlapping occurs between the arrival times, the sampling approach can result in multiple sequences. As an example, Figure 4-5 shows 4 possible sequences that can be obtained for the same entities if the uniform sampling approach is applied. The figure highlights entity number 2 for illustration purposes. The same entity replaces different positions in the event list based on its sampled instance w.r.t. the other entities. This is due to uncertainty modeling in simulation.

S_1	2	1	3	4	5	6	7	8	11	10
S_2	3	1	7	4	2	6	8	12	10	9
S_3	1	2	3	4	6	8	7	13	11	12
S_4	2	1	3	5	7	9	13	8	11	10

Front — Back

Figure 4-5: 4 different sequences of the first 10 customers in an [M]/[M]/1 system based on the uniform sampled approach

The time sampled approach to advance the simulation clock does not follow the FIFO basis at either the lower or the upper events occurrence times. However, it follows the FIFO discipline based on the sampled instances from the uniform distribution. All entities reserve a place in the event list according to the sampled instant from the interval times.

4.2.3.1 Comparison between SOMC and the IBS Uniform Sampled Approach

If we run the IBS using this clock advancement approach, the event list is created based on real variates that are uniformly sampled. Hence, the IBS estimates here are real-point values as compared to the two previous approaches and may be complementary to the SOMC. Used in conjunction with SOMC, it provides real estimates of performance measures accounting for the total uncertainty.

In SOMC (Vose, 2000), an analyst uses a probability distribution to model distributions' parameters, i.e. $X \sim \exp(\text{Uniform}(a, b))$. For each run, the analyst samples a parameter value from its distribution, i.e. $\lambda = a + u(b - a)$ where $u \sim \text{Uniform}(0,1)$. Consequently, each run is based on one single value of the parameter. Therefore, the SOMC simulation claims by definition that there is no uncertainty involved in a single run. In simpler words, the analyst is forced to either eliminate the imprecision or ignore it. Moreover, SOMC can lead to a high computational burden due to the number of iterations needed to achieve a solution which is as close as possible to the complete solution. Often, a Monte Carlo analyst is not aware of the number of replications required to achieve a certain level of robustness. This opens them up to significant criticism (Tucker & Ferson, 2003).

In this approach, the IBS analyst asks for only the minimum and maximum values of distribution's parameters represented as intervals, i.e. $X \sim \exp\left(\left[\underline{\lambda}, \overline{\lambda}\right]\right)$. In the IBS run, the lower and the upper bounds of the input distributions are generated. These bounds enclose all SOMC distributions resulted from parameter sampling. Regardless, the IBS, following this clock advancement approach, is executed based on uniformly sampled

instances from the lower and the upper bounds. Therefore, each simulation run is not based on a single parameter value but every random variate throughout a single run is generated from a different parameter, i.e. $\lambda \sim \text{Uniform}\left(\underline{\lambda}, \overline{\lambda}\right)$.

In addition, by not expressing the imprecision in the simulation results using this uniform sampling approach, we still need to run the IBS multiple times as opposed to the lower and the upper-based sorting. In uniform sampling approach of the IBS, we still do not provide an answer to the number of replications needed. However, the imprecise input distributions used in the IBS signal a certain level of system uncertainty by its parameters' widths. This can be a potential topic of future research to relate the uncertainty in the input parameters with the required number of IBS replications to achieve a certain measure of robustness. When the imprecision is large, it dictates larger number of replications to be constructed, and vice versa.

The differences between the IBS with the uniform sampling approach and the SOMC simulation mechanisms are shown in Figure 4-8.
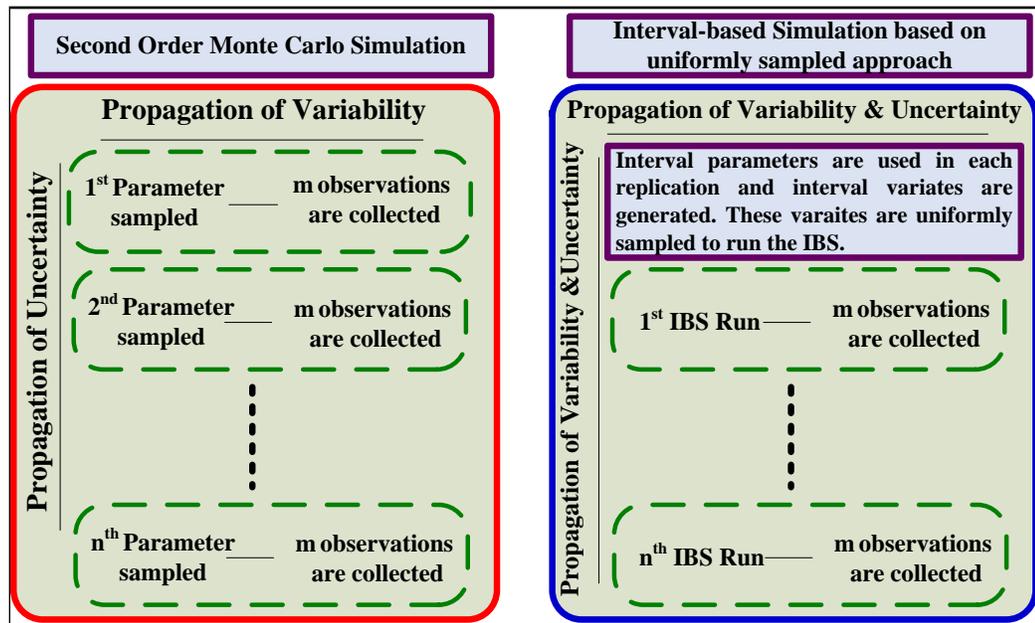
Figure 4-6: IBS with uniformly sampled approach to advance the simulation clock vs. SOMC

The uncertainty in the SOMC is propagated only across the different runs as a single parameter value is used to run one SOMC replication. The variability is represented in all the replications using the distribution functions as illustrated in the right diagram of Figure 4-7In the IBS, the variability and the uncertainty are modeled in each replication and across them. The uncertainty is modeled in the two directions because different values of the parameters are used to run a single IBS run. The explicit inclusion of uncertainty in a single run is obvious as each sample is observed from a different parameter value. Moreover, the variability is also modeled by the statistical distributions for the different replications. On the other hand, the variability is modeled by a different distribution at each IBS run. The uniform sampling approach produces a new distribution which is unnecessarily same as the input distribution. The developed distributions are

also enclosed within the lower and the upper bounds as shown in the left diagram of Figure 4-7.
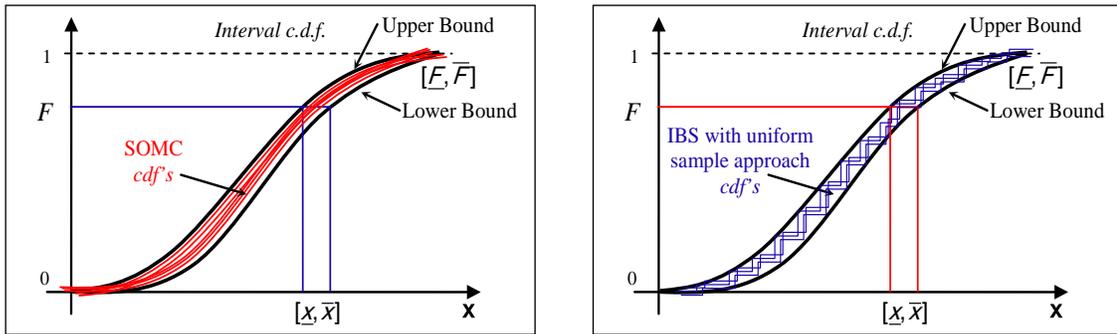
Figure 4-7: SOMC replications vs. IBS uniform sampled approach enclosed within the IBS bounds

More formally, the IBS bounds shown in the right diagram of Figure 4-7 enclose precise *cdf*'s obtained from the SOMC runs. On the other hand, the IBS bounds enclose imprecise *cdf*'s obtained from the IBS runs as demonstrated in the left diagram of Figure 4-7. If we think of the IBS uniform sampling approach and SOMC as drivers to reliable decision making, which approach would we prefer to adopt? Because the uncertainty in the IBS is explicitly modeled in each run and across the different runs, thus, the IBS is considered more reliable than the SOMC.

The question that arises here: is whether running two traditional simulation experiments using the lower and the upper parameters at each run estimates the best and worst-case scenario of the performance measures of interest. As a consequence, we estimate the performance measures using an easy and traditional approach. We answer this question by referring to our main objective of developing the IBS. The goal is to

model the input uncertainties in each simulation run and ultimately estimate the best and the worst-case scenarios. Simply, running a traditional simulation using the extreme bounds does not propagate the uncertainty component in the simulation. Our intention of running the DES is to account for the input uncertainties in each single run.

4.3   Output Analysis

The outputs of the interval-based simulation mechanism are also intervals. Appropriate statistical measures must be developed to help interpret simulation results as well as to design and analyze simulation experiments to support decision making.

The interval variance defined in Section 2.3.2 to measure data dispersion is computationally expensive and impractical in a simulation with hundreds of thousands samples. We investigate new measures that serve the purpose of measuring simulation variances and uncertainties and at the same time are easy to compute. Preliminary research proposed three measures: (1) Data disparity, (2) Data Range, and (3) Nonspecificity, as follows.

4.3.1   Data Disparity

Data Disparity measures the variability of the data from its mean. Suppose that we have n random intervals $[\underline{x}_i, \overline{x}_i](i = 1, ..., n)$. The mean $[\underline{\mu}, \overline{\mu}]$ is calculated as in Eq.(1). Data disparity, which measures the dispersion of the interval data away from the mean, is calculated as $D = \sum_i^n d_i^2/(n-1)$, where d_i is the maximum dispersion of the ith interval $[\underline{x}_i, \overline{x}_i]$ and calculated in six different ways, depending on how $[\underline{x}_i, \overline{x}_i]$ is located w.r.t. the

mean, as illustrated in Figure 4-8. The computation of data disparity D requires a linear execution time $O(n)$.
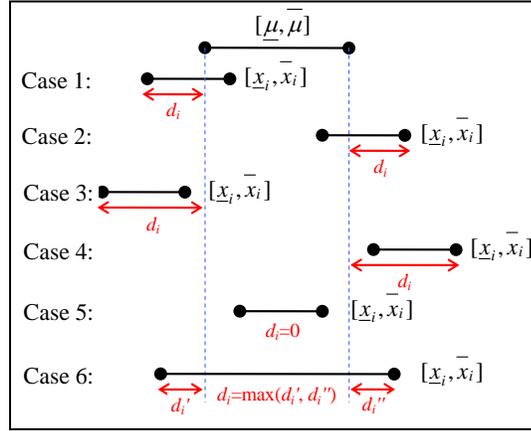
Figure 4-8: Six cases of data disparity d_i

4.3.2 Data Range

Data Range is to measure the level of uncertainty in estimations, which is calculated as the width of interval, $r_i = wid([\underline{x}_i, \overline{x}_i]) = |\overline{x}_i - \underline{x}_i|$ for $(i = 1, ..., n)$. The data range for a collection of random intervals can be calculated as either *mean* $\overline{r} = (1/n) \sum_{i=1}^{n} r_i$, *medium*, or *mode* of all widths.

4.3.3 Nonspecificity

Nonspecificity is also used to measure the level of uncertainty. Similar to the extended Hartley measure (Klir, 2006), it is calculated as $s_i = \log(1 + wid([\underline{x}_i, \overline{x}_i]))$ for the interval $[\underline{x}_i, \overline{x}_i]$. The collected nonspecificity is $S = \sum_{i=1}^{n} \log(1 + wid([\underline{x}_i, \overline{x}_i]))$.

The above described measures are easy to implement to assess interval data dispersion with respect to variability and uncertainty. Future research is needed to investigate these measures as explained in Chapter 6.

4.4   IBS Implementation using *JSim*

As our testbed, a library of Java-based interval DES toolkits, *JSim*, has been developed. The testbed is used to demonstrate the proposed new reliable simulation. The implementation of *JSim* involves all DES components in a next-event time-advance approach programmed in Java.

The following components are developed in *JSim* package to execute the IBS:

- *Source Object:* An object that generates entities in a system based on interval distributions. For instance, an entity (i) arrives to the system at time $[\underline{a}_i, \overline{a}_i]$.

- *Client Object:* An object that represents an entity generated in a system. They are created at time $[\underline{a}_i, \overline{a}_i]$, served for an interval time of $[\underline{s}_i, \overline{s}_i]$, and finally disposed at time $[\underline{d}_i, \overline{d}_i]$. These objects can be modified as convenient during the simulation.

- *Server Object:* A station that serves entities in a system during the simulation. Service times are based on interval distributions. As an example, station (j) provides a service to entity (i) for an interval time of $[\underline{s}_{ij}, \overline{s}_{ij}]$.

- *Sink Object:* An object that represents the exit of a system that disposes entity (i) after its service ends at time $[\underline{d}_i, \overline{d}_i]$.

- *Simulation Object:* This object determines the simulated system features. For example, the number of *Sources, Servers,* an*d Sinks* is specified in this object. In addition, the statistical distributions of random variables are identified here. In this object, the simulation clock is initiated at time $[0,0]$ and along with the clock initiation, the event list to handle the simulation is created. This event

75

list contains the interval times of the events and is modeled in *JSim* as an object called *Calendar* which is essentially a *priority queue.* In this chapter, we discussed three possible approaches to the interval events in the *Calendar.* In brief, the three approaches can be based on:

1) Lower event time: the events are sorted here based on the earliest possible time of occurrence, i.e. $\underline{a}_i$.

2) Upper event time: conversely, the events are sorted in the event list based on their latest possible time of occurrence, i.e. $\overline{a}_i$.

3) Uniform sampled event time: lastly, a suggested approach is based on a uniformly sampled time from the interval events, assuming that the interval time is uniformly distributed, i.e. $a \sim U(\underline{a}_i, \overline{a}_i)$.

The three proposed approaches can be implemented in *JSim* by changing the calendar prioritization rule as desired. *JSim* package is considered the basic platform to execute the IBS. Any extension to new objects can be built and added to the package. Object-oriented languages, like Java, are flexible to allow for customized models of different applications.

A bank example shown in Figure 4-9 is an illustration of modeling in *JSim*. It is modeled by one *Source,* one *Server,* and one *Sink.* In the simulation class, the inter-arrival and service times are defined as exponential distributions with interval parameters. In addition, a new *Calendar* is generated and is set to time [0,0]. Figure 4-10 shows a screen shot of [M]/[M]/1 implementation in *JSim* package using NetBeans Java editor.

76

Bank JSim Overview

BankPane

#control Pane: JPanel
#settingPane: JPanel
#viewPane: JPanel
#buttonStart: JButton
#buttonStats: JButton
#buttonStop: JButton
#tfStopTime: JTextFile
#btnSet: JButton
- thread: Thread=null

<<create>>+BankPane(): BankPane
+run(): void
+main(arg: String[]): void

BankServer

~queueID: int

<<create>>+BankServer(calendar: Calendar, serviceTime: IConRandom): BankServer
+onEvent(evt: SimulationEvent): void
#scheduleDeparture(client: EntityClient): void

~bankSim

~teller

BankSimulation

- thread: Thread=null
~stopTime: float
- status: int
+ST_STOP: int=0
+ST_RUN: int=1
+ST_PAUSE: int=2

<<create>>+BankSimulation(): BankSimulation
+getCalendar(): Calendar
+getSource(): BankSource
+getStopTime(): float
+setStopTime(newTime: float): void
+getStatus(): int
+setStatus(newStatus: int): void
+initialize(): void
+setProgressMonitor(progMonitor: ProgressMonitor): void
+start(): void
+pause(): void
+stop(): void
+run(): void
+generateStatistics(): void
++main(argv. String[]): void

~bankSink

BankSink

~name: String
- ID: int

<<create>>+BankSinkcalendar(): BankSink
+onEvent(evt: SimulationEvent): void

~bankSource

BankSource

~idCounter: int

<<create>>+BankSource(calendar: Calendar, InterArrivalTime: IConRandom): BankSource
+initialize(): void
+onEvent(evt: SimulationEvent): void
+setInterArrivalTime(random: IConRandom): void
- scheduleArrival(): void

~bankClient

BankClient

<<create>>+BankClient(timeOfArrival: float): BankClient

Figure 4-9: [M]/[M]/1 Bank Example Overview
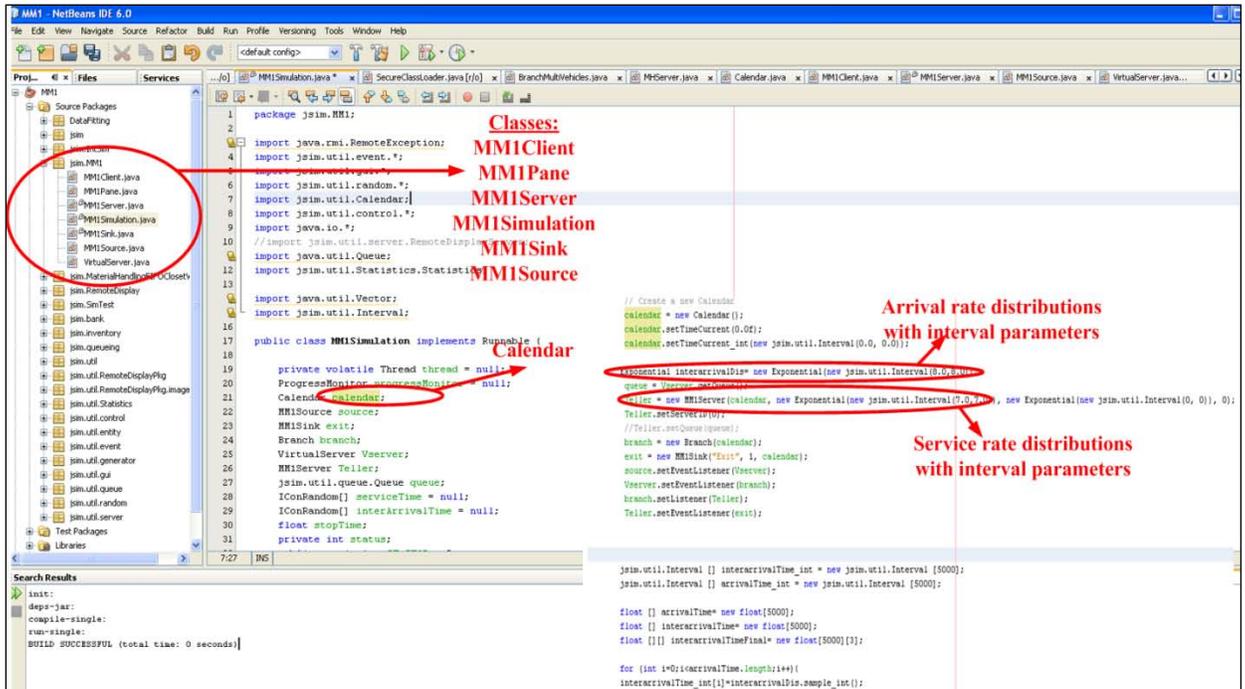
Figure 4-10: Screen shot from [M]/[M]/1 Implementation in NetBeans

4.5 The [M]/[M]/1 example in the IBS

The objective of this section is to formulate [M]/[M]/1 and illustrate the IBS mechanism using this example.

4.5.1 The automaton Model for [M]/[M]/1

The automaton model for this system is outlined as following:

- The set of events is: $\mathbb{E} = \left\{ \{\underline{a}, \overline{a}\}, \{\underline{a}, \overline{d}\}, \{\underline{d}, \overline{a}\}, \{\underline{d}, \overline{d}\}, \{\underline{a}, \emptyset\}, \{\emptyset, \overline{a}\}, \{\underline{d}, \emptyset\}, \{\emptyset, \overline{d}\} \right\}$,

  where $\underline{a}$ and $\overline{a}$ denote an arrival event based on the lower and upper inter-arrival bounds, respectively, and where $\underline{d}$ and $\overline{d}$ denote a departure event based on the

78

lower and upper departure bounds, respectively. And the symbol ϕ is used to indicate a void event occurred either at the lower or the upper bound. We define $\{\underline{x}, \overline{x}\} | \underline{x} \in \mathbb{Z}, \overline{x} \in \mathbb{Z}$ to represent a generalized discrete interval. A generalized interval, as mentioned earlier, is not constrained by $\underline{x} \leq \overline{x}$ any more.

- The state space is $\mathcal{S} = \{\{0,0\}, \{0,1\}, \{1,0\}, \{1,2\}, \dots \{3,2\}, \dots\}$ which represents the number of customers in the queue based on the simulation of the lower and the upper bounds.

- The set of feasible events is defined as $\Gamma^{-1}: \mathcal{S} \mapsto \mathbb{E}$. The different $\Gamma(\{\underline{x}, \overline{x}\}) = \left\{\{\underline{a}, \overline{a}\}, \{\underline{a}, \overline{d}\}, \{\underline{d}, \overline{a}\}, \{\underline{d}, \overline{d}\}, \{\underline{a}, \emptyset\}, \{\emptyset, \overline{a}\}, \{\underline{d}, \emptyset\}, \{\emptyset, \overline{d}\}\right\} \forall \left\{\{\underline{x}, \overline{x}\} \in \{\underline{X}, \overline{X}\}: \underline{x} > 0 \text{ and } \overline{x} > 0\right\}$ and $\Gamma(\{0,0\}) = \{\underline{a}, \emptyset\} \vee \{\underline{a}, \overline{a}\} \vee \{\emptyset, \overline{a}\}$, where the symbol $\vee$ is used to represent the logical disjunction (or).

- The state transition function $f: \mathcal{S} \times \mathbb{E} \mapsto \mathcal{S}$ has eight possible state transitions:

(1) $f(\{\underline{x}, \overline{x}\}, \{\underline{a}, \overline{a}\}) = \{\underline{x} + 1, \overline{x} + 1\}$ (5) $f(\{\underline{x}, \overline{x}\}, \{\underline{a}, \overline{d}\}) = \{\underline{x} + 1, \overline{x} - 1\}$

(2) $f(\{\underline{x}, \overline{x}\}, \{\underline{d}, \overline{a}\}) = \{\underline{x} - 1, \overline{x} + 1\}$ (6) $f(\{\underline{x}, \overline{x}\}, \{\underline{d}, \overline{d}\}) = \{\underline{x} - 1, \overline{x} - 1\}$

(3) $f(\{\underline{x}, \overline{x}\}, \{\underline{a}, \emptyset\}) = \{\underline{x} + 1, \overline{x}\}$ (7) $f(\{\underline{x}, \overline{x}\}, \{\emptyset, \overline{a}\}) = \{\underline{x}, \overline{x} - 1\}$

(4) $f(\{\underline{x}, \overline{x}\}, \{\underline{d}, \emptyset\}) = \{\underline{x} - 1, \overline{x}\}$ (8) $f(\{\underline{x}, \overline{x}\}, \{\emptyset, \overline{d}\}) = \{\underline{x}, \overline{x} - 1\}$

A state transition diagram for this system is shown in Figure 4-11. Note that the state space of this model is infinite (but countable). The hand simulation of the IBS for this example is discussed in the following section.
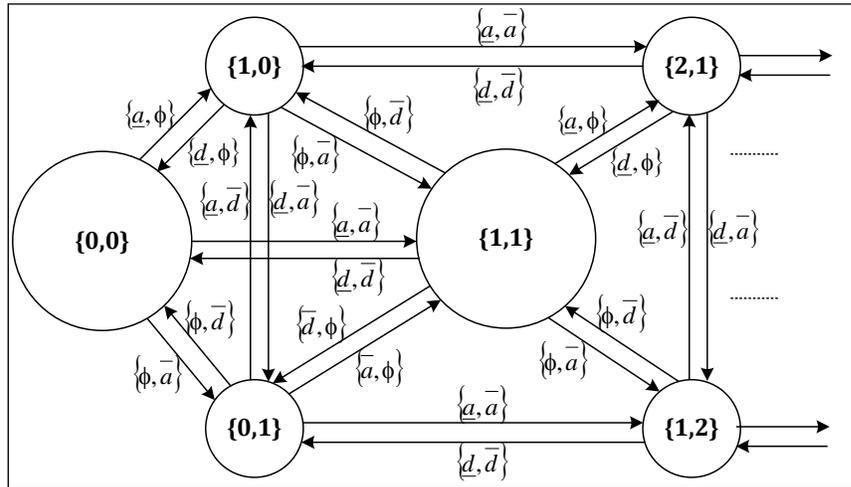
Figure 4-11: [M]/[M]/1 queueing system state transition diagram

4.5.2   Hand Simulation for [M]/[M]/1

In this section, we illustrate the IBS with an example of [M]/[M]/1. The purpose of this example is to show the execution of the IBS using a hand simulation, interpret the interval attributes and performance measures to verify completeness and soundness of numerical estimations, and estimate interval performance measures.

Example: Consider an operation of a single bank teller with an [M]/[M]/1 queue with the interval arrival rate of $[\underline{\lambda}, \overline{\lambda}] = [3,4]$ customers per hour and the interval service rate is $[\underline{\mu}, \overline{\mu}] = [5,5]$ customers per hour. Here the service rate is precise. The bank opens its doors at 9 A.M. and closes at 5 P.M., but it stays open until all customers at 5 P.M. have been served. The objective of the single-server queueing simulation is to estimate the expected interval steady-state sojourn time $[\underline{t}, \overline{t}]$ (i.e., time spent in the system).

80

For a stable system, the interval arrival rate is to be partially less than the interval service rate, i.e. $[\underline{\lambda}, \overline{\lambda}] \prec [\underline{\mu}, \overline{\mu}]$. The interval service-start time for entity (i) is defined as in Eq.(23) and the interval departure time $[\underline{d}_i, \overline{d}_i]$ is calculated as

$$[\underline{d}_i, \overline{d}_i] = [\underline{sst}_i, \overline{sst}_i] + [\underline{s}_i, \overline{s}_i] \tag{37}$$

where $[\underline{s}_i, \overline{s}_i]$ is the interval service time of entity (i). The interval arithmetic is used to calculate the sojourn time performance measure of entity (i) $[\underline{t}_i, \overline{t}_i]$ as

$$[\underline{t}_i, \overline{t}_i] = [\underline{d}_i, \overline{d}_i] - [\underline{a}_i, \overline{a}_i] \tag{38}$$

In Eq.(38), if the resulting lower bound is negative, it is set to the minimum service time associated with entity (i). We assume that the minimum possible time in the system is the time needed to complete the service.

Table 4-2 represents the hand simulation of the above example. Note that the simulation ends when the departure time upper bound $\overline{d}_i$ is approximately equal to *480 minutes*, assuming that the simulation's start time is [0,0]. The entity id i is given as in the first column. The event list formed by the lower and the upper-based sorting is similar. This is due to the existence of one *Source* generating the entities to the system. However, for more complex systems, the difference in the lower and upper based sorting is more obvious. The second and the third columns represent interval random variates of the arrival and service times, respectively. These random variates are generated as discussed in Section 3.2. Columns four, five and six represent the intervals of the service-start times, the departure times, and the sojourn times obtained from simulation, respectively.

81

TABLE 4-2
THE IBS HAND SIMULATION FOR THE [M]/[M]/1 EXAMPLE

Entity (i)	$[\underline{a}_i, \overline{a}_i]$ (min)	$[\underline{s}_i, \overline{s}_i]$ (min)	$[\underline{sst}_i, \overline{sst}_i]$ (min)	$[\underline{d}_i, \overline{d}_i]$ (min)	$[\underline{t}_i, \overline{t}_i]$ (min)
1	[15.82,21.09]	[5.73,5.73]	[15.82,21.09]	[21.54,26.82]	[5.73,11.00]
2	[30.72,40.96]	[73.85,73.85]	[30.72,40.96]	[104.57,114.81]	[73.85,84.09]
3	[61.00,81.33]	[10.78,10.78]	[104.57,104.57]	[115.35,115.35]	[34.02,54.35]
4	[111.80,149.07]	[4.80,4.80]	[115.35,149.07]	[120.15,153.87]	[4.80,42.07]
5	[121.77,162.36]	[18.96,18.96]	[121.77,162.36]	[140.73,181.33]	[18.96,59.55]
6	[139.83,186.45]	[14.15,14.15]	[140.73,186.45]	[154.89,200.60]	[14.15,60.77]
7	[141.64,188.85]	[58.77,58.77]	[154.89,188.85]	[213.66,247.62]	[58.77,105.98]
8	[147.14,196.19]	[28.61,28.61]	[213.66,213.66]	[242.27,242.27]	[46.08,95.13]
9	[164.27,219.03]	[6.83,6.83]	[242.27,242.27]	[249.09,249.09]	[30.06,84.82]
10	[172.57,230.09]	[12.49,12.49]	[249.09,249.09]	[261.59,261.59]	[31.50,89.02]
11	[181.87,242.49]	[0.04,0.04]	[261.59,261.59]	[261.62,261.62]	[19.13,79.75]
12	[186.55,248.73]	[3.81,3.81]	[261.62,261.62]	[265.43,265.43]	[16.70,78.88]
13	[228.16,304.21]	[0.42,0.42]	[265.43,304.21]	[265.85,304.63]	[0.42,76.47]
14	[231.05,308.07]	[18.01,18.01]	[265.85,308.07]	[283.86,326.08]	[18.01,95.03]
15	[232.17,309.56]	[4.94,4.94]	[283.86,309.56]	[288.80,314.50]	[4.94,82.33]
16	[272.98,363.98]	[4.69,4.69]	[288.80,363.98]	[293.49,368.67]	[4.69,95.69]
17	[288.20,384.27]	[13.04,13.04]	[293.49,384.27]	[306.54,397.31]	[13.04,109.11]
18	[288.82,385.09]	[1.80,1.80]	[306.54,385.09]	[308.33,386.89]	[1.80,98.07]
19	[294.51,392.67]	[16.91,16.91]	[308.33,392.67]	[325.25,409.59]	[16.91,115.08]
20	[297.82,397.09]	[3.07,3.07]	[325.25,397.09]	[328.32,400.16]	[3.07,102.35]
21	[312.91,417.22]	[9.80,9.80]	[328.32,417.22]	[338.12,427.02]	[9.80,114.11]
22	[320.08,426.78]	[0.77,0.77]	[338.12,426.78]	[338.89,427.54]	[0.77,107.46]
23	[384.85,513.13]	[7.23,7.23]	[384.85,513.13]	[392.08,520.37]	[7.23,135.52]

White cells are randomly generated values from Eq.(15)
Grey cells are calculated from Eqs.(23) (37)(38) respectively

The intervals of the entity attributes (i.e. arrival time, service time, and service-start time) and the performance measures (i.e. sojourn time) in the simulation are solely *proper* intervals. Based on Eq.(37), the quantified proposition for entity (i)

$$\left(\forall sst_i \in \left[\underline{sst}_i, \overline{sst}_i\right]\right)\left(\forall s_i \in \left[\underline{s}_i, \overline{s}_i\right]\right)\left(\exists d_i \in \left[\underline{d}_i, \overline{d}_i\right]\right)(d_i = sst_i + s_i) \qquad (39)$$

is true. From Eq.(38), the following quantified proposition for entity (i)

$$\left(\forall d_i \in \left[\underline{d}_i, \overline{d}_i\right]\right)\left(\forall a_i \in \left[\underline{a}_i, \overline{a}_i\right]\right)\left(\exists t_i \in \left[\underline{t}_i, \overline{t}_i\right]\right)(t_i = d_i - a_i) \qquad (40)$$

is also true. From Eq.(39) and Eq.(40) the combined quantified proposition

82

$$(\forall sst_i \in [\underline{sst_i}, \overline{sst_i}])(\forall s_i \in [\underline{s_i}, \overline{s_i}\,])(\forall a_i \in [\underline{a_i}, \overline{a_i}])(\exists d_i \in [\underline{d_i}, \overline{d_i}])(\exists t_i \in [\underline{t_i}, \overline{t_i}\,])(t_i = sst_i + s_i - a_i) \qquad (41)$$

is entailed. Therefore, the logic interpretation given in Eq.(41) helps to verify that the uncertainty estimation for the [M]/[M]/1 system is complete. That is, if the traditional simulation is used to simulate the M/M/1 system with any parameter values within the intervals, $\lambda \in [\underline{\lambda}, \overline{\lambda}]$ and $\mu \in [\underline{\mu}, \overline{\mu}]$, the resulted time in system for any entity, t_i, is always bounded by the IBS interval solution, i.e. $t_i \in [\underline{t_i}, \overline{t_i}\,]$. If we are interested in the average time in system for the n entities, the interval average time in system is calculated as in Eq.(1).

For two entities (i) and $(i + 1)$, the average of time in system $[\underline{t}, \overline{t}] = [(\underline{t_i} + \underline{t_{i+1}})/2, (\overline{t_i} + \overline{t_{i+1}})/2]$ is interpreted as

$$(\forall t_i \in [\underline{t_i}, \overline{t_i}])(\forall t_{i+1} \in [\underline{t_{i+1}}, \overline{t_{i+1}}\,])(\exists t \in [\underline{t}, \overline{t}])(t = (t_i + t_{i+1})/2) \qquad (42)$$

Combining Eq.(42) with Eq.(41), we can assert that the average value estimate is complete for two entities. This can be easily extended to the average estimate of n entities as in Eq.(1). Therefore, the estimate from the traditional simulation of average time in system t is always bounded by the interval estimate if the arrival and service rates are bounded in their associated intervals. For this example, the average interval time in system is $[\underline{t}, \overline{t}] = [0.31, 1.67]$ hours.

In this example, if we calculate a different sojourn time performance measure $[\underline{t'_i}, \overline{t'_i}]$ of entity (i) using the *dual* operator as in

$$[\underline{t'_i}, \overline{t'_i}] = [\underline{d_i}, \overline{d_i}] - dual[\underline{a_i}, \overline{a_i}] \qquad (43)$$

This solution will be sound but not necessarily complete. The *dual* operator in Eq.(43) is introduced in the Kaucher arithmetic. Compared to the *semi-group* formed by the classical set-based intervals, generalized intervals form a *group*. Therefore, the addition of the interval arrival time and the interval sojourn time calculated based on Eq.(43) is always equal to the interval departure time as in

$$[\underline{a}_i, \overline{a}_i] + [\underline{t}'_i, \overline{t}'_i] = [\underline{d}_i, \overline{d}_i] \tag{44}$$

whereas $[\underline{t}_i, \overline{t}_i]$ calculated from Eq.(38) does not have this property. Eq.(43) is interpreted as

$$\left(\forall a_i \in [\underline{a}_i, \overline{a}_i]\right)\left(\forall t_i \in [\underline{t}'_i, \overline{t}'_i]\right)\left(\exists d_i \in [\underline{d}_i, \overline{d}_i]\right)(a_i + t_i = d_i) \tag{45}$$

Eq.(37) can also be interpreted as

$$\left(\forall d_i \in [\underline{d}_i, \overline{d}_i]\right)\left(\exists sst_i \in [\underline{sst}_i, \overline{sst}_i]\right)\left(\exists s_i \in [\underline{s}_i, \overline{s}_i]\right)(d_i = sst_i + s_i) \tag{46}$$

since all $[\underline{d}_i, \overline{d}_i]$'s, $[\underline{sst}_i, \overline{sst}_i]$'s and $[\underline{s}_i, \overline{s}_i]$ are proper intervals. From Eq.(45) and Eq.(46), the combined quantified proposition is

$$\left(\forall t_i \in [\underline{t}'_i, \overline{t}'_i]\right)\left(\forall a_i \in [\underline{a}_i, \overline{a}_i]\right)\left(\exists sst_i \in [\underline{sst}_i, \overline{sst}_i]\right)\left(\exists s_i \in [\underline{s}_i, \overline{s}_i]\right)\left(\exists d_i \in [\underline{d}_i, \overline{d}_i]\right)(t_i = sst_i + s_i - a_i) \tag{47}$$

for entity (i). The interpretation verifies that $[\underline{t}'_i, \overline{t}'_i]$ is a sound solution. In this numerical example, the average of the sound solutions for sojourn times based on Eq.(1) is $[\underline{t}', \overline{t}'] =$ [0.55,0.72] hours, which is tighter than the complete solution. However, the average is just a complete estimate of the sound individual sojourn times for all entities.

CHAPTER 5:
AN INTERVAL-BASED METAMODELING APPROACH TO SIMULATE MATERIAL HANDLING IN SEMICONDUCTOR WAFER FABS

This chapter discusses an interval-based metamodel for Automated Material Handling Simulation (AMHS) in comparison to traditional simulation models that are based on a detailed description of AMHS operations. The metamodel is based on the IBS in which the statistical distribution parameters in simulation are intervals instead of precise real numbers. The remainder of this chapter is organized as follows: Section 5.1 introduces the semiconductor manufacturing, description of the automated handling system and its modeling in *JSim* is illustrated in Section 5.2, and Section 5.3 summarizes its simulation inputs and outputs. Section 5.4 represents the metamodel validation by testing how its outputs closely resemble the output data of the detailed simulation. In addition, we compare the results of *JSim* metamodel to the AutoMod metamodel.

5.1 Semiconductor Manufacturing

Semiconductor technology, used in most modern electronics, is the building block of our information technology. The semiconductor industry is a vital contributor to the world economy, with $248.6 billion in sales worldwide in 2008, as reported by the Semiconductor Industry Association (SIA) pressroom (SIA report, 2009). The transition from 200mm to 300mm, and the potential transition to 450mm wafer fabrication is a key element of continuing productivity gains in semiconductor device manufacturing and is driving fabs towards the full automation of material flow. Automated Material Handling

Systems (AMHS's) are responsible for moving materials between production equipment and storage units to complete the processing of the wafers.

Constructing a 300mm fab costs $2-3 billion (Jones, 2003), while a 450mm fab is projected to cost $10 billion (LaPedus, 2009). The AMHS represents 3 to 5% of the total fab cost (Arzt & Bulcke, 1999). For the AMHS to have acceptable Return on Investment (ROI) and provide the expected support to the production equipment, efficient design and operational strategies must be investigated and tested in the design and re-design stages of the factory. An improperly designed or operated AMHS may introduce lot delays (increasing manufacturing cycle times) or cause tool idle time (reducing throughput or requiring excess capacity).

In recent years, particular attention has been given to the development of efficient design and operational strategies for wafer fabs. These efficient strategies must target increasing the throughput of the AMHS substantially with reduced delivery times. Further, the AMHS needs to be flexible and scalable to achieve the demands of the ever-changing semiconductor wafer fab.

Estimating AMHS performance in wafer fabs is difficult, because of the complexity of the systems. The International Technology Roadmap for Semiconductors (ITRS, 2007) characterizes the AMHS as having several vehicles, operating on a network with loops, intersections, spurs, and shortcuts, serving many different pick-up/drop-off stations. The movement requests appear to be random, and although they exhibit some temporal correlations, these correlations are not strong enough to permit precise scheduling of the AMHS resources.

A typical 300mm AMHS has a spine layout configuration, as illustrated in Figure 5-1. Most wafer fabs use this bay layout (Cardarelli & Pelagagge, 1995), where each bay contains a group of similar process tools. A spine layout consists of a central material handling spine (interbay) and loops branching on both sides (intrabays) to serve production equipment (tools) in the bays. Automated storage units, referred to as stockers, are used to provide temporary buffering for work-in-process.
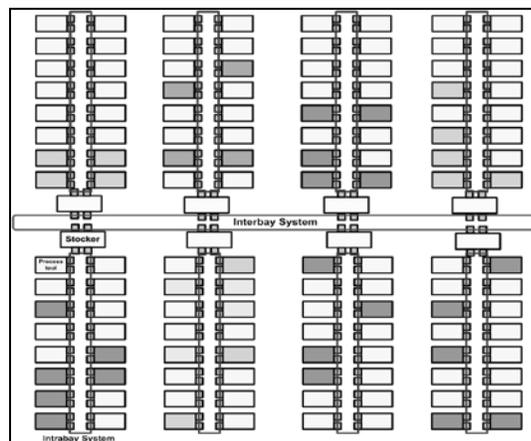
Figure 5-1: An AMHS in a spine layout - one interbay and 8 intrabay systems (based on ITRS 2005)

Almost all existing 300mm AMHS's are based on Overhead Hoist Vehicles (OHV) – space efficient vehicles traveling suspended on tracks above the main fab floor. The efficiency of an OHV-based AMHS is highly dependent on the vehicles' characteristics and control mechanism (i.e., speed, acceleration/deceleration, dispatching rules, etc.). An AMHS with a small number of vehicles will cause long delays for lots waiting to be transported. Clearly, longer wait times imply longer delivery times. On the other hand, an excess of vehicles can cause traffic congestion in the interbay and intrabay

systems because each of these units will frequently block other transporters that are traveling on the same path. As a result, delivery times increase significantly due to the longer delays that wafers experience while traveling in these highly congested systems.

Interaction between fab design (e.g., where to locate tools) and AMHS design (e.g. track configuration, fleet size, etc.) can have significant impacts on fab performance. Thus, the number of design alternatives for the AMHS is vast. Relying solely on discrete event simulation to navigate the AMHS design space means a commitment to a lengthy and expensive process, which may limit the range and number of alternatives that can be considered in the early stages of fab design. Simulation is ineffective as a decision support tool in the early phase of system design, where many configurations need to be considered. Our metamodeling approach, proposed and tested, is to simultaneously estimate accurate performance measures with shorter simulation time and incorporate input uncertainties in its estimations.

5.2   AMHS Metamodel based on the IBS

In the AMHS, sources of uncertainties could be due to vehicle congestion and blocking, vehicle and equipment breakdowns, and insufficient sample data to estimate systems random variables such as inter-arrival and service times. In other words, modeling these uncertainties in AMHS gives more reliable simulation results as their *completeness* and *soundness* with respect to uncertainties can be verified.

AMHS metamodel is an abstraction of the detailed simulation model. In our implementation, we represent the exact process routes by a number of move requests and their routing probability obtained from the detailed simulation. The general layout of the

example used to represent the AMHS is composed of 24 machines; 48 stations (24 loading and 24 unloading stations). This layout is based on a 300mm virtual semiconductor fabrication facility developed and published by International SEMATECH (SEMATECH, 2001). The vehicles travel on a uni-directional closed-loop at a constant speed of 3ft/sec. The product family modeled is SEMATECH's 300mm aluminum process flow for 180nm technology. Such technology nodes contain six metal layers and 21 masks. For this single product family, ten products are continuously released into the process. The release rate is 20,000 wafers per month (wpm). The processing route consists of approximately 316 operations (i.e. steps). In addition, there are 60 different workstations and about 300 tools. Wafers travel in carriers (lots) that hold 25 units. The 300mm Wafer Fab Model has 24 bays arranged using a spine layout configuration similar to the layout previously shown in Figure 5-1. We will only model the central aisle, also referred to as the interbay AMHS that transfers the wafers between the 24 bays. A schematic of the interbay system is shown in Figure 5-2.
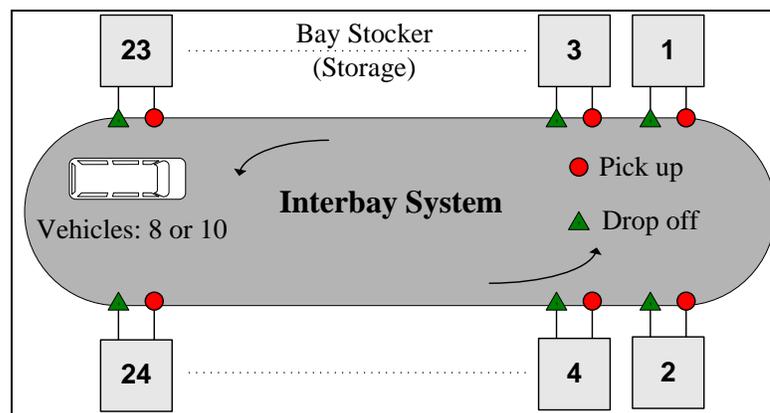
Figure 5-2: Schematic of the Modeled Interbay Systemc

89

The software used for detailed simulation is AutoMod 11.1. In order to obtain steady-state estimates, we start with an empty system and warm it up until it reaches steady state as indicated by the steady level of work-in-process in the system. After the warm-up period, all the appropriate statistics are collected. We refer to the results obtained from this simulation as "Detailed Results" because this simulation model explicitly models the wafers movement between the different bays, and these are assumed to be accurate estimates.

The IBS metamodel is implemented in *JSim*. A *Source* object is used to represent the bays that generate the *Entities (group of wafers, also referred to as lots)* with determined inter-arrival times. A *Server* is used to characterize the vehicles that transfer *Entities* in the interbay system. Finally, exits in the system are represented using a *Sink* to dispose an entity upon the end of its service time. Figure 5-3 illustrates the object oriented modeling for AMHS in *JSim.* More practically, Figure 5-4 shows the implementation of the metamodel in *JSim* using NetBeans Java editor.

Essentially, the metamodel does not explicitly model the wafers flow through each bay in its process route. Instead, details concerning the processing of wafers are implicitly represented by the number of move requests received by the AMHS. These moves are summarized in the metamodel as "From-To" matrices such as the one shown in Table B-4 of Appendix B, which describes the rate of moves between two different bays of the fab. The From-To matrices are generated from the production volume and the process route of the products in SEMATECH's model. The metamodel results are referred to as "IBS Metamodel Results".
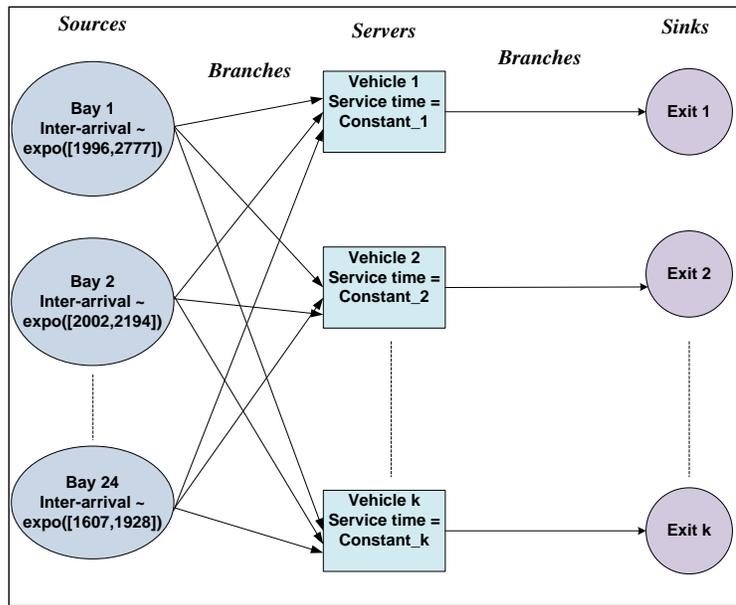
Figure 5-3: Object oriented modeling for AMHS implemented in *JSim*
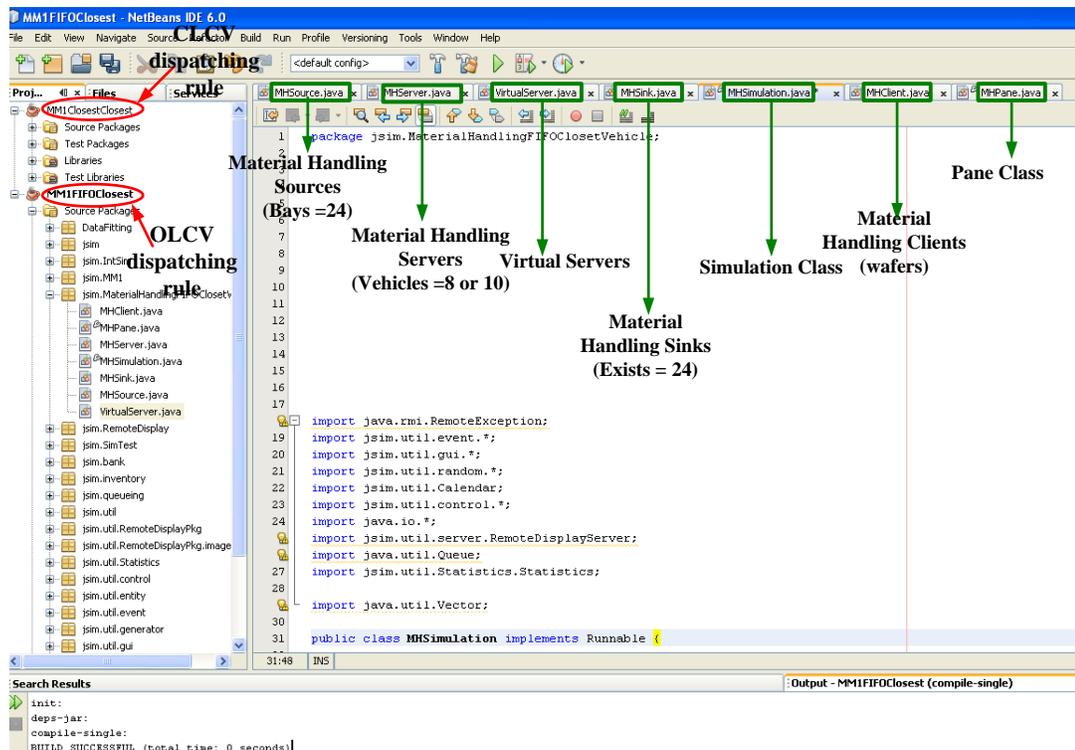
Figure 5-4: Screen shot of AMHS interval-based metatmodel implementation in JSim

5.3 Metamodel Simulation Process

5.3.1 Interval Input Random Variates

In this AMHS system, it is assumed that we do not have enough information to be certain about the parameters of the inter-arrival times for the bays. Only 1000 sample-points from the "Detailed Simulation" are collected to fit an exponential distribution with real-valued parameter β using the maximum likelihood estimator (MLE) for each bay, as in Table B-1 of Appendix B. Based on the obtained value from the MLE, the proposed interval-parameterization technique in Section 3.2 is used to find the interval mean of the exponential distributions. The order r is selected based on the obtained ratios of $\underline{\beta}/\beta$ and $\overline{\beta}/\beta$ at orders $r = 1,2,\dots 1000$, illustrated in Figure 5-5a and 5-5b.
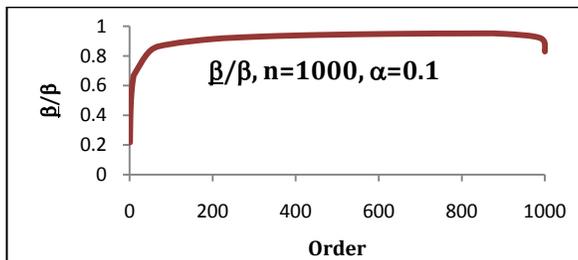
Figure 5-5a Ratio of $\underline{\beta}/\beta$ with order r Figure 5-5b Ratio of $\overline{\beta}/\beta$ with order r
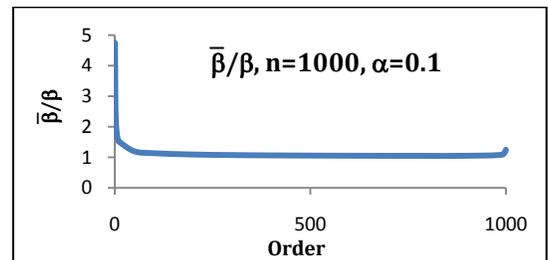
Due to the narrow width of the *cdf* bounds, i.e. the *cdf* curves become flatter at those bounds, it becomes more difficult to bound the real-valued variable at small and very large orders. If the interval parameter obtained at these orders is used, we tend to estimate a complete solution that includes all possible occurrences at all orders. For instance, the minimum of the lower bounds $\underline{\beta}$'s, and the maximum of the upper bounds

$\overline{\beta}$'s give the worst-case estimations. The ratios for the lower and the upper parameters to the real-valued mean are approximately the same for middle orders. The simulation analyst can choose the interval parameter associated with a particular (r) based on the desired level of enclosure. In our implementation, we use order $r = 100$ with a confidence level of $\alpha = 0.1$ to estimate the interval-parameters needed in simulation.

The obtained ratios of $\underline{\beta}/\beta$ and $\overline{\beta}/\beta$ are 0.8506 and 1.1831, respectively. Hence, we multiply these ratios by the real-valued mean obtained from the MLE for each bay to find the interval mean. For example, the real-valued average inter-arrival time obtained for the first bay is $\beta = 2347.40$ seconds. Hence, the corresponding interval mean at $r = 100$ is $\left[\underline{\beta}, \overline{\beta}\right] = [0.8506\beta, 1.1831\beta] = [1996.70, 2777.21]$. Table B-1 in Appendix B presents the inter-arrival times of entities of the 24 bays with the real-valued parameters and the associated intervals. Because order $r = 100$ is selected out of $n = 1000$, at least 90% enclosure of the ordered real-valued random variates between their corresponding interval variates is guaranteed. Moreover, a probability of at least $(1 - \alpha) = 90\%$ is guaranteed to enclose the real-valued variate between the bounds of interval variate at each order. For instance, if we run these bounds n times, we are confident that at least $n(1 - \alpha)$ times the interval variates enclose the real variates generated from the exponential distribution $\exp(\beta)$.

The obtained intervals for the inter-arrival times in Table B-1 are used to run the metamodel to enclose the detailed simulation results. Note that no entities are generated from bays 6 and 23. Additionally, service times are assumed to be constant to transfer entities between the bays. Table B-2 in Appendix lists the total service times, which are

the summations of the empty vehicle travel times from the vehicle location to the load location, and the loading times used in this simulation. The From-To routing probabilities matrix is also listed in Table B-3 of Appendix.

5.3.2 Metamodel Simulation Results

For this metamodel, we study two dispatching rules to serve the entities, 1) oldest load closest vehicle (OLCV), and 2) closest load closest vehicle (CLCV). The OLCV dispatching rule ensures that waiting entities are served based on a FIFO principle while selecting the closest idle vehicle to serve the entities. Similarly, the CLCV dispatching rule selects the closest vehicle to serve an entity. However, it serves the closest waiting entity when a vehicle becomes idle. The two scenarios are simulated varying the fleet size between 8 and 10 vehicles.

In the metamodel, entities are generated with interval arrival times, i.e. $[\underline{a}_i, \overline{a}_i]$ for entity (i). We decide the sequence of serving entities based on the upper bounds of their interval arrival times $\overline{a}_i$, i.e. the latest time the entities arrive at the system. The upper-based sorting estimates the worst-case scenario of the response time to move requests.

From the IBS metamodel, we are interested in calculating the interval response time to move requests. The response time to a move request, i.e. the waiting time in the queue, is the sum of the waiting time until a vehicle becomes idle and the travel time of empty vehicle to the load location. In addition, we study the enclosure of these intervals to the real-valued response time to move requests obtained from the detailed simulation. The interval response time to move requests is now calculated as

$$[\underline{w_i}, \overline{w_i}] = [\underline{sst_i}, \overline{sst_i}] - dual[\underline{a_i}, \overline{a_i}]$$ (48)

Eq.(48) gives a range estimate to the waiting time of the entities to be served. The *dual* operator is used to estimate a sound solution to the response time to move requests in comparison with the complete solution that results from the interval arithmetic without using the *dual* operator as in Eq.(49). The solution provided by Eq. (48) is a sound solution that does not include impossible solutions. Hence, some real-valued solutions may be out of the calculated bounds from Eq. (48).

But they all are bounded by the complete solution from

$$[\underline{w_i}, \overline{w_i}] = [\underline{sst_i}, \overline{sst_i}] - [\underline{a_i}, \overline{a_i}]$$ (49)

However, the complete solution usually overestimates and gives every wide bounds. In this IBS metamodel, we use Eq.(48) for calculation. Assume that entity (i) interval arrival time is given as $[12.34, 17.64]$ and the interval service-start time is given as $[18.45, 25.65]$ second. If Eq.(48) is used to estimate the interval response time, the solution is $[6.11, 8.01]$ second and its interpretation is

$$(\forall a_i \in [12.34, 17.64])(\forall w_i \in [6.11, 8.01]]) \ (\exists sst_i \in [18.45, 25.65])(sst_i - a_i = w_i) \quad (50)$$

However, if Eq.(49) is used to calculate the interval response time, the complete solution is $[0.81, 13.31]$ second and interpreted as follows

$$(\forall a_i \in [12.34, 17.64])(\forall sst_i \in [18.45, 25.65])(\exists w_i \in [0.81, 13.31])(sst_i - a_i = w_i) \quad (51)$$

Moreover, we represent the variation in the interval response time resulted by calculating the standard deviations for the lower bounds $\underline{w_i}$'s. In addition, the vehicles' utilization is measured by the percentage of time the vehicle is loaded, travels with entities, and unloaded. In the IBS metamodel, the vehicles' average utilizations are given

95

as real-valued estimates. The average utilization are calculated as the percentage of time the vehicles travel to serve an entity, regardless of the entity arrives at its lower bound $\underline{a}_i$, or its upper bound $\overline{a}_i$.

Such models are valuable to early stages of design because it allows the designer to experiment with different design strategies for the number of vehicles and the flow path layout. Increasing the number of vehicles has the potential to reduce the expected response time to move requests, which is directly related to the production cycle time of the wafers. Reducing the production cycle time is always a priority for fabs because of the short life span of these types of products. However, there is an optimal number of vehicles to install, beyond which the improvement in response time is marginal and may not be justifiable financially. Fab designers benefit from the metamodel as it provides fast answers to different design scenarios. The importance of monitoring the standard deviation of response times is two-fold: first, inconsistent response times translate to inconsistent delivery times to the end customer, an undesirable and expensive situation as increased variability is directly related to increased levels of safety stocks. Second, from simple queueing formulas, we know that increased variability propagates through a manufacturing line and increases the work-in-process and the queueing delays at subsequent stages.

In the *JSim* implemention of the IBS metamodel, we excecuted $n = 5$ independent replications for both OLCV and CLCV scenarios. The number of replications was selected so that the confidence intervals of the simulation outputs have a half-width to mean ratio of less than 5%. Each has a length of $m = 200$ days. Conservatively, we chose a warm-up period of $l = 100$ days to reach the steady-state.

96

The next two sections summarize the simulation results. The performance measures include the interval time to move request $[\underline{w_i}, \overline{w_i}]$, the standard deviation of the lower bounds $s(\underline{w_i})$, and the average utilization of the vehicles ρ, with respect to the two dispatching rules:

5.3.2.1 Oldest Load Closest Vehicle Rule

First, we present the results obtained from the OLCV dispatching rule with the simulation of 8 and 10 vehicles to transfer entities. The simulation results of the average response time using 8 vehicles are shown in Figure 5-6. We compare the lower and upper bounds obtained from the IBS with the detailed simulation results obtained from AutoMod.
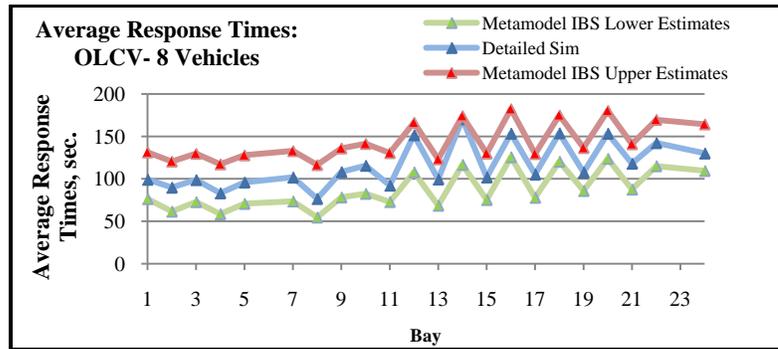
Figure 5-6: Average response times to move requests for OLCV – 8 Vehicles

The lower and the upper estimates of the interval results enclose the detailed simulation results. Thus the uncertainties associated with the inter-arrival times of the entities at the bays are incorporated. For instance, we report the response time for bay 1 as [76.35,131.33] second as opposed to the detailed real-valued simulations that only

give an estimate of 81.64 second. The interval estimations of the performance measures are considered more reliable as it provides a range of solutions that enclose the detailed simulation results incorporating uncertainties in the inter-arrival times. The gap between the interval bounds and the detailed simulation results is due to modeling the uncertainty component in simulation, which is expected and desired. From the results, we notice that the differences between the bounds and the detailed simulation results are consistent for the different bays in the system.

One might ask how these intervals differ from the standard confidence intervals. We answer this concern by referring to these traditional methods as statistical measures that incorporate only the variability component in their estimates. For instance, the traditional confidence interval limits represent a lower and upper bounds of the estimates based on a marginal error in the readings with a certain level of confidence. The interval limits are calculated as the mean value of the outputs of multiple simulation runs +/- a quantity that represents the standard deviation in these outputs. The standard deviation is attained within these readings because of the different random number streams used in the simulation runs. Given that all simulation runs use a fixed value of the parameters, the interpretation of this interval is that the average mean of the performance measure is included between these interval limits with a certain level of confidence. They do not represent the uncertainty in their bounds. However, our interval estimates incorporate the variability and the uncertainty components explicitly in each single simulation run. The input distributions with imprecise parameters provide interval estimates to the performance measures of interest from each simulation run. Because the uncertainty is propagated in the simulation runs, our intervals are not a result of running multiple

simulation runs. Instead, they are obtained from running the IBS with imprecise parameters where uncertainty is incorporated within one single run. In addition, in traditional simulation output analysis, confidence intervals are indicators of the confounded effect of variability and uncertainty. In IBS, the effects of the two components are quantified separately and can be treated in different ways in decision making. Therefore, the IBS intervals results are considered more reliable than the traditional confidence intervals.

Furthermore, the standard deviations of the lower bound response times are collected and compared with those from the detailed simulation. Figure 5-7 depicts the difference in the standard deviations for the detailed and IBS metamodel with 8 vehicles. The standard deviations from the detailed simulation are larger than the ones obtained from the IBS metamodel. However, they both follow the same pattern for different bays. However, they both follow the same pattern for different bays. The lower standard deviation of the IBS metamodel is less than that from the detailed simulation model because it is calculated from the lower response times. Because of the equal values of mean and standard deviation in an exponential distribution, the lower response times resulted from the simulation of entities arriving at the system have a lower variability.

In addition, the average utilization of the 8 vehicles is reported as 56.66% for the detailed simulation, and 48.70% for the IBS metamodel. The difference is because the vehicle traveling times in the IBS metamodel, are averages of the actual ones in the detailed model without variations.
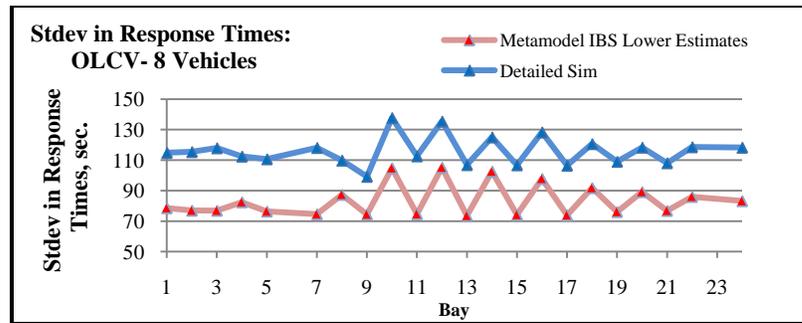
Figure 5-7: Standard deviation of response time to move requests for OLCV – 8 Vehicles

Figure 5-8 presents the response times to move requests with 10 vehicles. In this setting, the system becomes more saturated with vehicles and the response time decreases as the availability of the vehicles increases. The increase in availability of the vehicles reduces the uncertainty in the response times to move requests. In other words, the response times to move requests in such scenarios are less uncertain because there are more vehicles to serve the entities whether they arrive at their lower or upper arrival times. Therefore, the differences between the response times from the detailed simulation and the lower or upper bounds from the metamodel are small. The response times of the detailed simulation for all bays except bays 1, 8, 11, 14 and 24 are enclosed by the corresponding intervals from the IBS metamodel. When order r=100 was selected, we were aiming a 10% of enclosure for each bay separately not for all the bays together. This is interpreted as follows: the intervals means at each bay includes at least 90% of real-point means obtained from traditional simulation. Again, the lack of complete enclosure using 10 vehicles in simulation is because the system is more saturated with vehicles than it is needed.
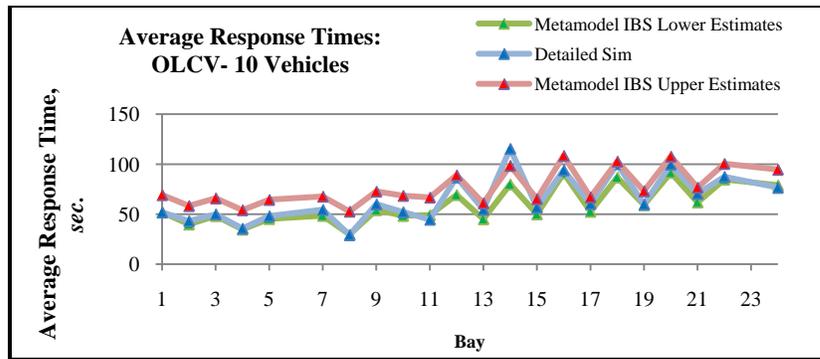
Figure 5-8: Response time to move requests for OLCV – 10 Vehicles

The standard deviations associated with the response times using 10 vehicles are illustrated in Figure 5-9. The standard deviations for response times in the detailed simulations are slightly greater than the standard deviations of lower bounds from the IBS metamodel. Again, both estimates follow the same pattern. The utilization of the 10 vehicles from the detailed simulation is given as 43.38% down to 39.94% for IBS metamodel.
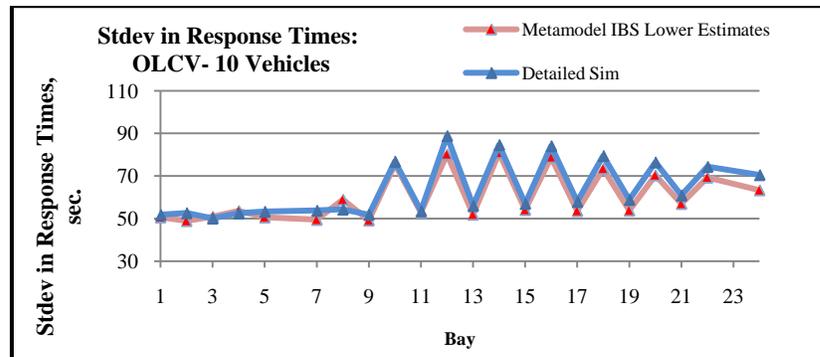
Figure 5-9: Standard Deviation in Response time to move requests for OLCV – 10 Vehicles

5.3.2.2 Closest Load Closest Vehicle Rule

We also model the AMHS using the CLCV dispatching rule for 8 and 10 vehicles. Figure 5-10 presents the average response times for 8 vehicles and the standard deviation of the response times for each bay. The average response times obtained from the detailed simulation are well-enclosed between the lower and the upper bounds obtained from the IBS metamodel.
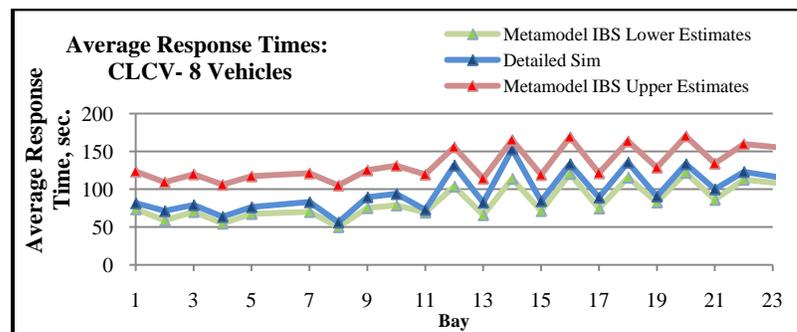
Figure 5-10: Response time to move requests for CLCV – 8 Vehicles
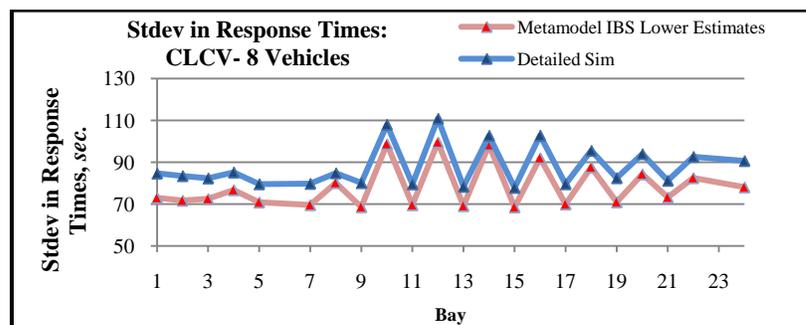
Figure 5-11: Standard Deviation in Response time to move requests for CLCV – 8 Vehicles

Figure 5-11 compares the standard deviations of the lower bounds from the metamodel and the ones from the detailed simulation. The average utilization of the vehicles for this scenario is reported as 54.60% for the detailed simulation and as 48.26% for the IBS metamodel.

As for 10 vehicles, the simulation results are summarized in Figure 5-12 and Figure 5-13. The average response times obtained from the detailed simulation model are not well enclosed within the bounds of IBS metamodel. As mentioned above, the reason is due to the increased number of vehicles. Hence, the average response time is comprised mostly of travel times of empty vehicles to the waiting entities. Vehicles are mostly available when a request is issued. In addition, the standard deviations of the two simulations are quite close to each other with at most 16.28% of relative differences. The average utilizations of vehicles are 41.47% for the detailed simulation and 35.62% for IBS metamodel. There is no relationship noticed between the selected dispatching rule and the enclosure of the IBS results to the detailed simulations outputs. The enclosure of the IBS to the detailed simulation is shown for most of the bays regardless of the dispatching rule.
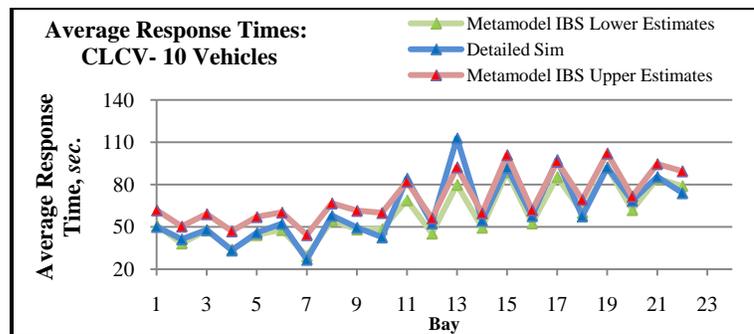
Figure 5-12: Response time to move requests for CLCV – 10 Vehicles
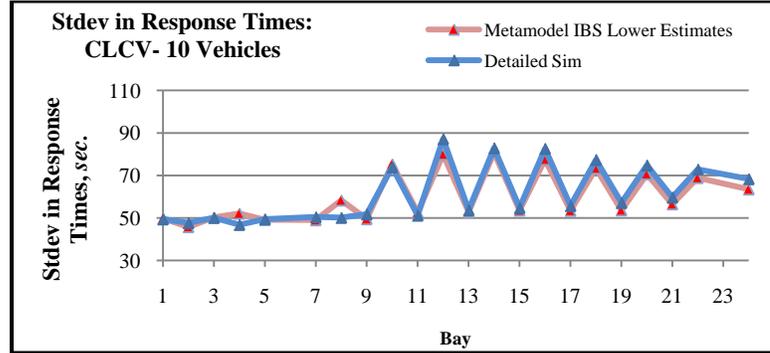
Figure 5-13: Standard Deviation in Response time to move requests for CLCV – 10 Vehicles

In summary, the IBS metamodel offers interval estimations for average response times enclosing the detailed simulations with certain level of confidence. Moreover, the interval estimations model input uncertainties in the interarrival times of entities. The input uncertainties come from unknown dependency between bays, machine breakdown, and vehicle congestion. The standard deviations obtained from lower bounds follow the same pattern as the detailed simulation variations. However, the IBS standard deviations are less than the corresponding results obtained from the detailed simulations. The vehicles' utilizations calculated from IBS are also smaller than the detailed simulation estimates.

The simulation time based on the metamodel is significantly reduced. On a dual-processor workstation, one run of the IBS model takes less than 2 minutes, whereas the detailed simulation requires 30 minutes on average.

5.4   Validation of the *JSim* Metamodel

In this Section, we compare *JSim* metamodel's results to the AutoMod model of the SEMATECH virtual fab when both models are using from-to matrices rather than the detailed process flow of the products. The objective of this analysis is to validate our metamodel with respect to the detailed simulation when using precise parameters for the inter-arrival times. In addition, this analysis compares the *JSim* metamodel to the AutoMod metamodel. The motivation for this is that often in the literature, for the purpose of modeling the material handling systems, from-to matrices are used rather than an explicit representation of the production system.

We run the metamodel using AutoMod and its results are referred to as "AutoMod Metamodel Results". Moreover, we use the interval-based *JSim* platform replacing the interval parameters with real-point parameters and compare the results. A number of independent replications $n = 5$ are executed for the two metamodels and for the dispatching rules, OLCV and CLCV. Each replication has a length of $m = 200$ days, and a warm-up period of $l = 100$ days to reach the steady-state.

We assume that the detailed simulation results are desired threshold which we compare the metamodel results to. Let μ be the average output of the detailed simulation results, which is the acceptable surrogate for actual performance. Let $\overline{X}$ be the average output of the AutoMod and the IBS metamodels results. We use the relative error to assess the dispersion between the detailed and the metamodel results. The relative error is calculated as the absolute difference in the statistics collected $\overline{X}$ and μ, i.e.

$Relative\ error = \frac{(\overline{X} - \mu)}{\mu} \times 100\%$. The following sections display the relative error of the two dispatching rules mentioned above.

5.4.1 Oldest Load Closest Vehicle Rule

Figure 5-14 depicts the relative error in the average response time of AutoMod metamodel and IBS metamodel with respect to the detailed simulation for OLCV with 8 vehicles.
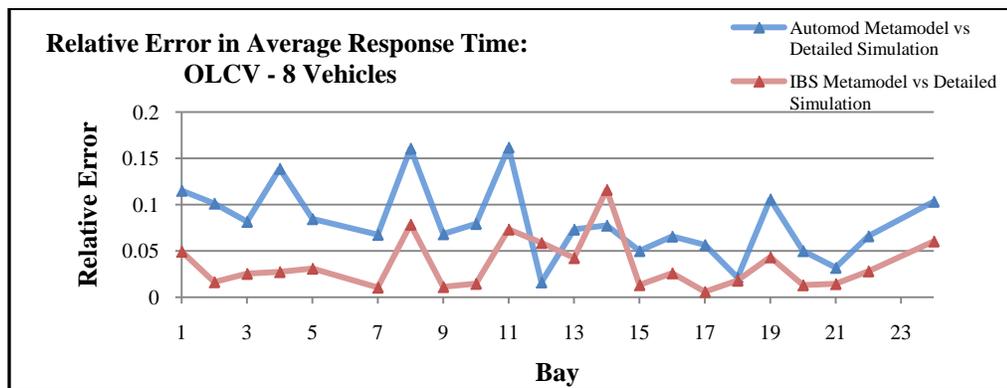
Figure 5-14: Relative Error in Average Response time to move requests for
OLCV – 8 Vehicles

The relative error given by IBS metamodel is always less than the associated error reported from the AutoMod metamodel. Hence, IBS metamodel results are closer to the detailed simulation than the AutoMod results. Hence, the IBS metamodel with precise input parameters is considered more reliable than the Automod metamodel. Figure 5-15 studies the relative error in the standard deviations in response times for the two metamodels with respect to the detailed simulation.
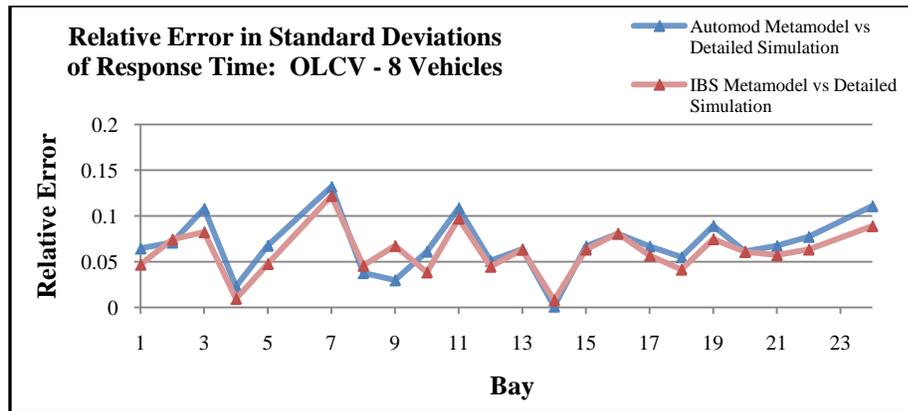
Figure 5-15: Relative Error in Standard Deviations in Response time to move requests for

OLCV – 8 Vehicles

The relative errors in the standard deviations given by the metamodels are quite close to each other. In reference to Figure 5-15, the difference is practically insignificant. Finally, the relative errors in the average utilization for the two metamodels with respect to the detailed simulation are reported as 6.88% and 7.52% for AutoMod metamodel and the IBS metamodel, respectively. The relative errors are quite close to each other.

Below are the same validation analyses performed for OLCV rule with 10 vehicles. Figure 5-16 shows the relative error in the average response time of OLCV rule using a fleet size of 10 vehicles. The relative error produced from using IBS metamodel is less than the associated error from using AutoMod metamodel for most of the bays. The relative errors in the standard deviations are demonstrated in Figure 5-17.
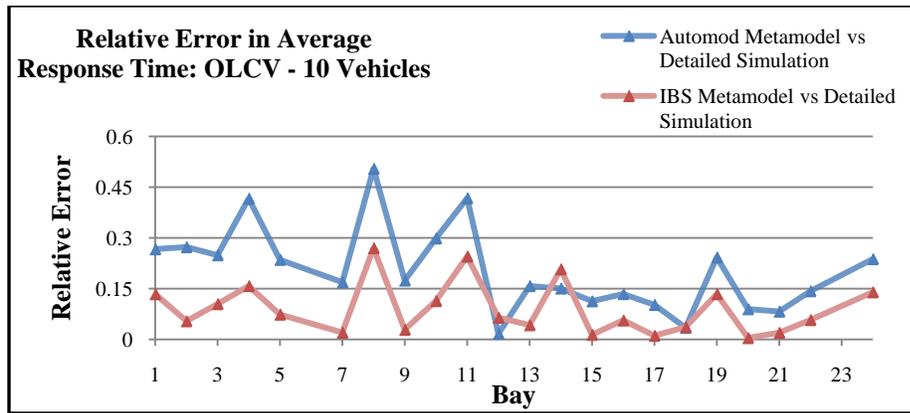
Figure 5-16: Relative Error in Average Response time to move requests for
OLCV – 10 Vehicles
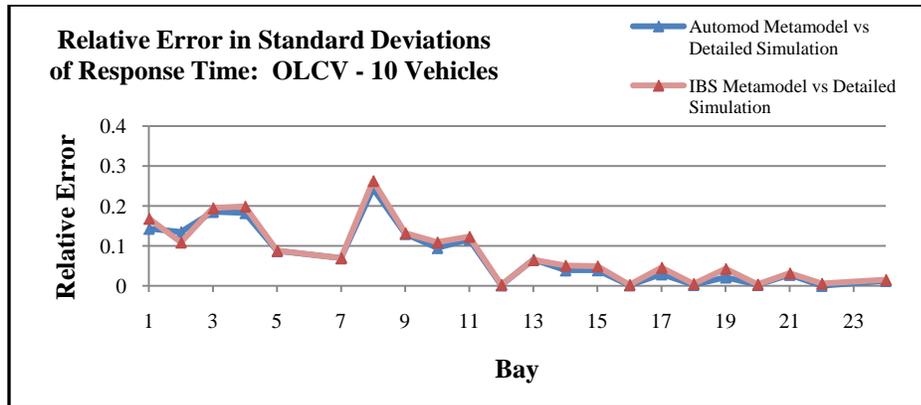
Figure 5-17: Relative Error in Standard Deviations in Response time to move requests for
OLCV – 10 Vehicles

Practically speaking, the difference in the standard deviations in the response times given by the metamodels is insignificant. Looking at the average utilization of the vehicles, we obtain a relative error of 8.16% for AutoMod and 8.61% for IBS metamodel. Both metamodels give almost equal average utilizations of the vehicles. For this

dispatching rule, we notice that the IBS metamodel performed better than the AutoMod metamodel in terms of the average response times.

5.4.2 Closest Load Closest Vehicle Rule

The same validation analyses are performed for the closest load closest vehicle rule using 8 and 10 vehicles to transfer entities. Figure 5-18 shows the relative error in the average response time for the CLCV rule using 8 vehicles as the fleet size. IBS metamodel offers better results than AutoMod metamodel with respect to average response times. As shown in Figure 5-19, the errors standard deviations in response times resulted from metamodels with respect to detailed simulation are almost equal to each other.
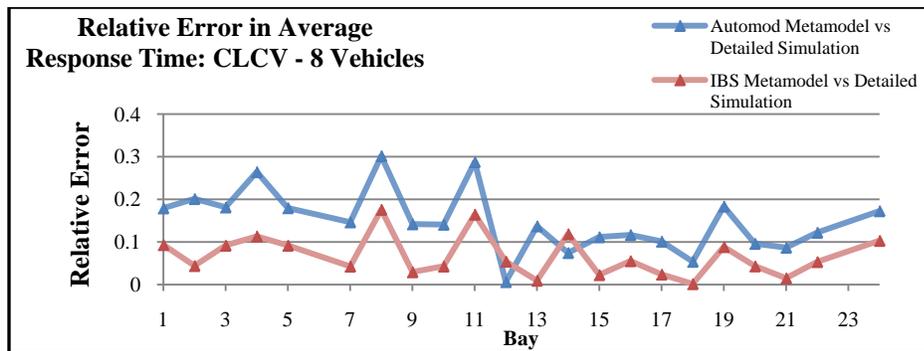
Figure 5-18: Relative Error in Average Response time to move requests for CLCV – 8 Vehicles
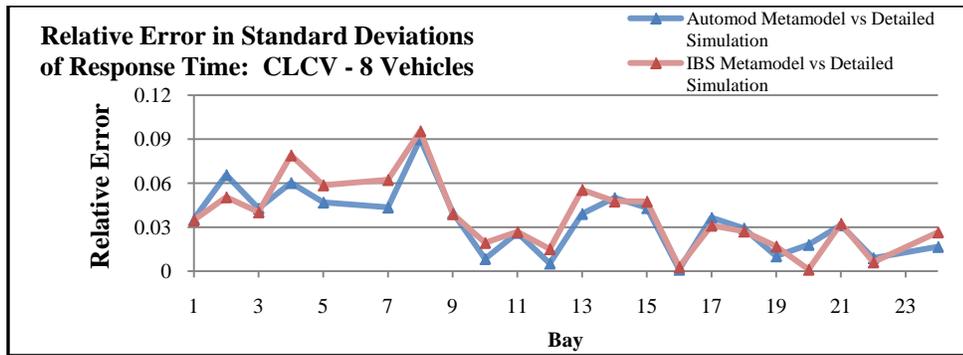
Figure 5-19: Relative Error in Standard Deviations in Response time to move requests for
CLCV – 8 Vehicles

As a final performance measure, the error in the average utilization is calculated
as 7.95% and 8.17% for AutoMod and IBS metamodels, respectively, with respect to the
detailed simulation. Finally, we present the validation analysis for the CLCV dispatching
rule using 10 vehicles, and Figure 5-20 represents the relative error in the average
response time for this scenario. Additionally, IBS metamodel performs better than the
AutoMod metamodel for this CLCV dispatching rule.
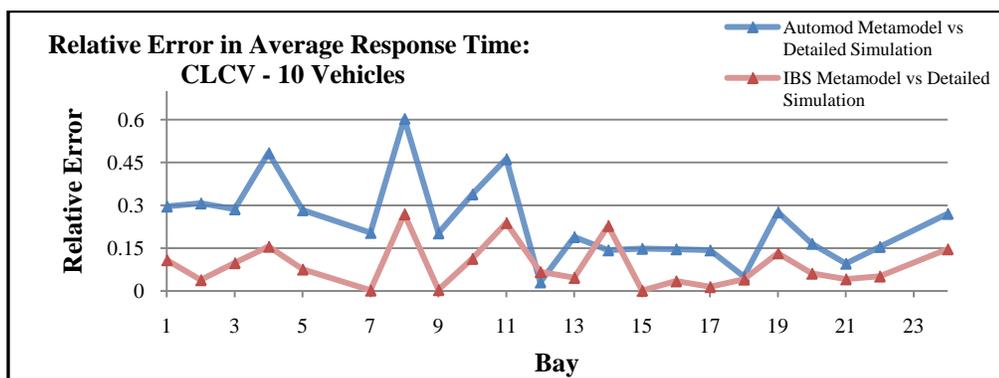
Figure 5-20: Relative Error in Average Response time to move requests for
CLCV – 8 Vehicles

The relative errors in the standard deviation resulted for this scenario using AutoMod and IBS metamodels are shown in Figure 5-21. We can notice that the relative errors in the standard deviations using either metamodel are equal with respect to the detailed simulation. Additionally, the relative error in the average utilization comparing the metamodels to the detailed simulation is reported as 9.17% for AutoMod metamodel down to 5.15% running IBS metamodel. In this scenario,The IBS metamodel wins over the Automod metamodel with respect to the vehicles utilization.
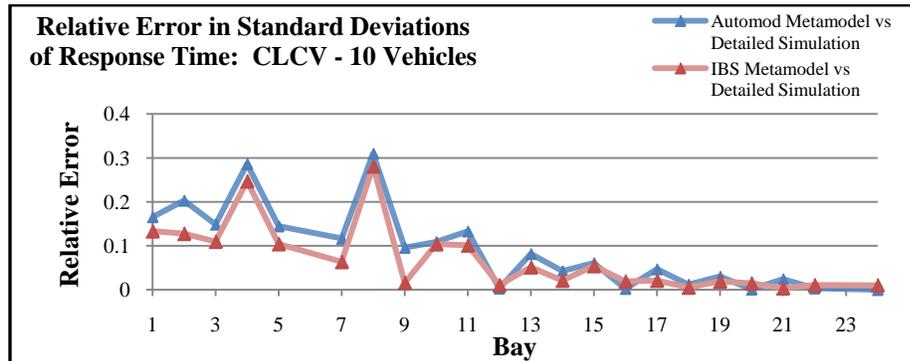
Figure 5-21: Relative Error in Standard Deviations in Response time to move requests for CLCV – 10 Vehicles

The errors are quite small in most occurrences and IBS metamodel still gives better results than AutoMod metamodel. Hence, *JSim* is more reliable to be used for running the metamodel with real-point parameters than using AutoMod software. Lastly, we conclude that our metamodel models the detailed simulation in a right manner.

111

CHAPTER 6:
SUMMARY AND FUTURE RESEARCH WORK

6.1   Summary

Uncertainty quantification in simulation notably increases the robustness of the DES to support decision making. The work presented in this dissertation exploits the interval-based simulation to address the total uncertainty in simulation. In practice, particularly for new designs of queueing systems without data, simulation analysts usually have to turn to an ad hoc choice of the parameter values. Moreover, they usually choose larger or smaller parameters than necessary to avoid design failure. This frequently results in costly designs. In the IBS, the simulation analysts do not need to resort to an ad hoc approach to determine a precise value of the parameters. Conversely, the incompletely known values are modeled imprecisely using intervals. This approach results into imprecise simulation events accounting for the uncertainty in the system and driving to a more reliable and less costly solution compared to the traditional practices.

Here, we propose a new framework of DES to model input uncertainties in simulation where:

1) Statistical distributions with interval input parameters are used to represent the uncertainties, i.e. $\exp([\underline{\lambda}, \overline{\lambda}])$. Consequently, lower and upper bounds are built to enclose the real-valued distributions. The interval parameters are used as an attempt to wrap all the uncertainty factors in the system. The IBS can be viewed as a generalized DES which allows the modeling of both precise and imprecise distributions. For example, if the process owners are certain about their service rate to

be distributed as $\exp(\mu)$. This complete knowledge in the system transforms the imprecise probability into a precise one and is modeled in the IBS as $\exp([\mu, \mu])$.

2) An interval parameterization technique is developed to enclose all real-valued distributions with a certain level of confidence. A concrete way to model this problem is based on order statistics sampling distributions. This technique has been implemented on single parameter distributions, i.e. exponential distribution, as well as on multiple parameter distributions, i.e. normal distribution. The parameterization technique can be applied to any distribution with a known form.

3) A simulation robustness measure is studied to find the number of replication needed to run the IBS for a given interval parameters to enclose all real-valued distributions with a certain level of confidence. The robustness measure is derived for the exponential, normal, and triangular distribution which is also based on order statistics sampling distribution. The robustness measure is applicable to any distribution with a known form.

4) An interval variate generation technique is explained based on inverse transform for single and multiple parameter distributions. The exponential distribution is used to illustrate the method regarding single parameter distributions. On the other hand, multiple parameter distributions generate $2^{\# \, of \, parameters}$ possible combinations that generate multiple random variates at each order. The interval variate is built based on the minimum and the maximum values of the random variates. The normal and triangular distributions are used to demonstrate the idea.

5) A new interval-based simulation clock is investigated and three approaches to advance the simulation clock in the IBS are proposed. With these proposed

113

approaches demonstrated, it is essential that the analyst understands the system and defines the problem properly before selecting the clock advancement approach. Table 6-1 compares the three proposed approaches.

TABLE 6-1: COMPARISON OF THE THREE PROPOSED APPROACHES TO ADVANCE THE SIMULATION CLOCK IN THE IBS

Difference	Lower-based Sorting	Upper-based Sorting	Uniformly Sampled-based Sorting
Sorting Criteria	Earliest possible occurrence time, i.e. $\underline{a_i}$.	Latest possible occurrence time, i.e. $\overline{a}_i$.	Uniformly sampled occurrence time, i.e. $a_i \sim \text{Uniform}(\underline{a_i}, \overline{a}_i)$
Scenario estimated	Imprecise best-case scenario	Imprecise worst-case scenario	Average-case scenarios
Simulation Results	Interval form	Interval form	Real-valued results as opposed SOMC

The formulation of the lower and upper-based sorting allows us to recognize an imprecise best and worst case scenarios w.r.t. performance measures of interest, respectively. Whereas, the uniformly sampling approach represents a middle ground between the extreme scenarios and provides real-valued estimates that yet account for input uncertainties.

6) Because the data is now represented as intervals, the analyst cannot use the statistical measures in the traditional sense. Instead, new statistical measures of variability and uncertainty are required. Three dispersion statistical measures are proposed to quantify the variability and the uncertainty of interval data in a simple approach and less computational time. Three proposed measures are data disparity, data range, and nonspecificity.

114

7) Most simulation analysts are familiar with real-valued analysis, but many have not been formally exposed to the competing interpretation of imprecise probabilities. We provided the logical interpretation of the hand simulation results for the [M]/[M]/1 example. The philosophical arguments for the logical interpretations can be quite passionate to verify their soundness and completeness.

8) The IBS is implemented to address the automated material handling simulation. This example shows that the IBS is applicable to broader class of problem than the [M]/[M]/1 queueing systems. We propose an AMHS metamodel based on the IBS to simulate a 300m wafer fab. The parameters of probability distributions for the inter-arrival times in the simulation are intervals instead of traditional precise numbers. We implement the metamodel in *JSim*. The obtained interval estimates to the mean response time are considered more reliable compared to the real-valued estimates, since it incorporates the total uncertainty in simulation. Experimental comparisons indicate that the IBS metamodel performs very well for estimating the average and standard deviation of response times at each bay. They are critical performance measures when evaluating the AMHS. Our numerical results also show that the metamodel enclosure of the detailed simulation results deteriorates as the servers (vehicles in this case) are under-utilized. This is expected because as the AMHS becomes less congested, variability in its performance reduces and the advantage of interval performance measures becomes less obvious.

9) We also validate the *JSim* metamodel with precise input parameters by assessing the relative error between its results and the detailed simulation. We also conclude that

the *JSim* metamodel gives better results than the AutoMod simulation outputs when they are compared to the detailed simulation results.

6.2   Future Work

Some potential ***future research work*** to extend the interval-based simulation involves:

1) Input Analysis:

   a. Statistical distribution fitting and parameterization: optimization procedures are potential for research on the completeness and soundness in selecting distributions types and parameters.

   b. Empirical distribution functions with interval parameters could be investigated as input distributions in the IBS.

   c. Interval variate methods as acceptance rejection method could be employed with interval parameters.

2) Simulation Clock: The three proposed approaches to advance the simulation clock could be analyzed in a rigorous and quantitative manner to verify its simulation results. Based on the adopted approach to handle the events, generalized intervals (i.e. proper and improper) of the simulation results can be obtained. Such results should be properly explored and interpreted to study their effects on the decision making process.

Other approaches for event handling could be based on the intervals length, based on intervals overlapping cases, or even a simple id based on a certain criteria. Each approach estimates different best and worst-case scenarios. The IBS analyst selects the appropriate approach based on the target application.

3) Output Analysis: New statistics for interval data incorporating the total uncertainties are to be developed to help support robust decision making. Further research is needed to study the dispersion associated with the interval data obtained from the IBS. Studying the properties (subadditivity, range, continuity, expansibility, monotonicity, etc.) of the proposed measures is necessary to support decision making. For the AMHS proposed metamodel, we use the standard deviation of lower bounds to measure the dispersion of intervals. An interval-based statistics will be more reliable. This can help us to understand more simulation details thus support robust decision making in layout selection.

Moreover, the confidence level measures w.r.t. uncertainty similar to confidence interval w.r.t. variability is an area of research. It could be based on the data disparity D and the variance of data range $R = (\sum_{i=1}^{n}(r_i - \bar{r})^2)/(n-1)$. The new confidence level measures are useful in alternative systems comparisons based on the total uncertainty.

4) Improper intervals: Introducing improper intervals in the IBS provides richer interpretations for the simulation results. Improper intervals can be used to reduce output uncertainties. For the AMHS meatmodel, improper intervals could be used to model possible buffers in the system that can reduce request time to move requests. For example, if an entity (i) service-start time is given as a $\left[\underline{sst_i}, \overline{sst_i}\right] = [4,5]$ and its service time is given as proper interval of $\left[\underline{s_i}, \overline{s_i}\right] = [2,3]$ then the entity's departure time $\left[\underline{d_i}, \overline{d_i}\right]$ is calculated as

$$\left[\underline{d_i}, \overline{d_i}\right] = [4,5] + [2,3] = [6,8]$$

The uncertainty of the entity's departure time can be quantified to be equal to $wid(d_i) = |8 - 6| = 2$. However, if the service time is given an improper interval of $[\underline{s}_i, \overline{s}_i] = [3,2]$ as buffer that serves faster than in normal conditions then the entity's departure time is calculated as

$$[\underline{d}_i, \overline{d}_i] = [4,5] + [3,2] = [7,7]$$

The uncertainty within this departure time is calculated as $wid(d_i) = |7 - 7| = 0$. Thus the uncertainty is reduced. More research is required to investigate improper intervals in the IBS as an uncertainty reduction technique.

APPENDIX A CALCULATION OF THE IBS REPLICATION LENGTH NEEDED FOR AN EXPONENTIAL DISTRIBUTION

The replication length as a robustness measure is performed for an exponential distribution with interval ratio of $\left[\underline{\beta}/\beta, \overline{\beta}/\beta\right]$ as [0.9,1.1], [0.8,1.2] , and [0.6,1.4], and the results are summarized in Tables A-1, A-2 and A-3, respectively.

TABLE A-1: $\left[\underline{B}/B, \overline{B}/B\right] = [0.9,1.1]$

r	α=0.1
1	>1030 max(p)=0.271
2	>1030 max(p)=0.300
3	>1030 max(p)=0.317
4	>1030 max(p)=0.330
5	>1030 max(p)=0.342
6	>1030 max(p)=0.352
7	>1030 max(p)=0.362
8	>1030 max(p)=0.371
9	>1030 max(p)=0.379
10	>1030 max(p)=0.387
20	>1030 max(p)=0.452
30	>1030 max(p)=0.501
40	>1030 max(p)=0.542
50	>1030 max(p)=0.578
60	>1030 max(p)=0.609
70	>1030 max(p)=0.638
80	>1030 max(p)=0.664
90	>1030 max(p)=0.687
100	>1030 max(p)=0.708
110	>1030 max(p)=0.728
120	>1030 max(p)=0.746
130	>1030 max(p)=0.762
140	>1030 max(p)=0.777
150	>1030 max(p)=0.792
160	>1030 max(p)=0.804
170	>1030max(p)=0.817
180	>1030 max(p)=0.828
190	>1030 max(p)=0.838
200	>1030 max(p)=0.848
250	>1030 max(p)=0.888
300	433
350	418
400	448
450	488

TABLE A-2: $\left[\underline{B}/B\,,\overline{B}/B\right] = [0.8,1.2]$

r	α=0.1	α=0.05
1	>1030 max(p)=0.306	>1030 max(p)=0.306
2	>1030 max(p)=0.360	>1030 max(P)=0.360
3	>1030 max(p)=0.395	>1030 max(p)=0.395
4	>1030 max(p)=0.424	>1030 max(p)=0.424
5	>1030 max(p)=0.448	>1030max(p)=0.448
6	>1030 max(p)=0.470	>1030max(p)=0.470
7	>1030 max(p)=0.491	>1030 max(p)=0.491
8	>1030 max(p)=0.509	>1030max(p)=0.509
9	>1030 max(p)=0.526	>1030max(p)=0.526
10	>1030 max(p)=0.542	>1030max(p)=0.542
20	>1030 max(p)=0.666	>1030max(p)=0.666
30	>1030 max(p)=0.748	>1030max(p)=0.748
40	>1030 max(p)=0.806	>1030 max(p)=0.806
50	>1030max(p)=0.849	>1030 max(p)=0.849
60	>1030 max(p)=0.882	>1030 max(p)=0.882
70	143	>1030 max(p)=0.906
80	105	>1030 max(p)=0.926
90	106	>1030 max(p)=0.940
100	112	273
110	120	161
120	129	152
130	138	154
140	147	160

TABLE A-3: $\left[\underline{B}/B\,,\overline{B}/B\right] = [0.6,1.4]$

r	α=0.1	α=0.05
1	>1030 *max(p)*=0.396	>1030 *max(p)*=0.396
2	>1030 *max(p)*=0.506	>1030 *max(p)*=0.506
3	>1030 *max(p)*=0.577	>1030 *max(p)*=0.577
4	>1030 *max(p)*=0.631	>1030 *max(p)*=0.631
5	>1030 *max(p)*=0.675	>1030 *max(p)*=0.675
6	>1030 *max(p)*=0.712	>1030 *max(p)*=0.712
7	>1030 *max(p)*=0.743	>1030 *max(p)*=0.743
8	>1030 *max(p)*=0.769	>1030 *max(p)*=0.769
9	>1030 *max(p)*=0.792	>1030 *max(p)*=0.792
10	>1030 *max(p)*=0.811	>1030 *max(p)*=0.811
20	29	>1030 *max(p)*=0.920
30	32	46
40	42	45
50	51	53
60	61	62
70	71	72
80	81	82
90	91	91

APPENDIX B AMHS METAMODEL INPUT DATA

TABLE B-1: INTER-ARRIVAL TIMES REAL-VALUED EXPONENTIAL MEANS AND INTERVALS (*SEC.*)

Bay	Real-valued Mean (*sec.*)	Interval Mean (*sec.*)
1	2347.40	[1996.70,2777.21]
2	2002.39	[1703.23,2369.03]
3	1164.78	[990.76,1378.05]
4	1270.86	[1080.99,1503.55]
5	1170.68	[995.78,1385.03]
6	0	[0,0]
7	2345.79	[1995.33,2775.31]
8	303.92	[258.51,359.56]
9	3496.66	[2974.26,4136.89]
10	563.48	[479.29,666.65]
11	881.46	[749.77,1042.85]
12	1280.50	[1089.19,1514.96]
13	1407.37	[1197.11,1665.06]
14	1401.75	[1192.33,1658.42]
15	1396.91	[1188.21,1652.69]
16	2332.05	[1983.64,2759.05]
17	2331.11	[1982.84,2757.94]
18	2791.37	[2374.34,3302.47]
19	4675.92	[3977.33,5532.08]
20	7061.44	[6006.46,8354.39]
21	1993.41	[1695.60,2358.41]
22	2327.75	[1979.98,2753.96]
23	0	[0,0]
24	1759.17	[1496.35,2081.28]

TABLE B-2: TOTAL TRAVEL, LOADING AND UNLOADING TIMES NEEDED TO TRANSFER ENTITIES BETWEEN THE BAYS (*SEC.*)

From / Bay	To / Bay 1	2	3	4	5	6	7	8	9	10	11	12
1	0.00	318.25	14.49	304.91	27.81	297.31	41.10	284.06	54.44	270.65	67.76	257.33
2	21.08	0.00	34.40	324.82	47.72	317.21	61.01	619.92	74.35	593.12	187.40	277.24
3	325.12	304.92	0.00	291.58	14.48	283.98	27.77	553.15	41.11	257.32	54.43	244.00
4	34.48	14.27	47.80	0.00	61.11	330.61	74.41	646.66	87.75	619.86	101.07	290.64
5	311.76	291.56	661.88	278.21	0.00	270.61	14.41	257.36	27.75	499.68	41.07	230.64
6	0.00	22.01	55.54	8.67	68.85	0.00	82.15	325.10	95.49	311.69	108.81	298.38
7	298.44	278.24	311.76	264.90	661.95	257.29	0.00	244.04	14.43	230.64	27.75	217.32
8	55.53	35.33	149.41	55.91	82.17	14.39	95.47	0.00	229.39	661.77	122.13	635.16
9	285.10	264.90	608.59	251.56	311.74	243.95	325.03	230.70	0.00	217.29	14.41	203.98
10	149.56	48.72	82.24	35.38	95.56	27.77	229.50	14.52	122.19	0.00	282.84	661.95
11	271.76	251.56	285.08	238.22	608.61	230.61	635.23	217.36	325.03	419.73	0.00	393.11
12	82.22	62.01	95.54	109.32	108.85	41.07	122.15	67.41	135.49	14.41	309.45	0.00
13	258.42	238.22	271.74	224.88	285.06	217.27	298.35	204.02	311.69	393.04	325.01	177.30
14	95.60	162.83	108.92	0.00	122.24	54.45	135.53	41.20	148.88	27.80	336.23	14.48
15	245.06	0.00	258.38	211.51	271.69	203.91	284.99	393.16	298.33	177.25	311.65	163.93
16	108.87	88.67	122.19	75.33	282.82	67.73	0.00	54.48	0.00	41.07	0.00	27.75
17	231.76	211.56	245.08	198.22	258.40	190.61	271.69	366.50	285.03	163.95	298.35	150.64
18	127.86	227.53	141.18	94.32	154.50	86.71	167.79	73.46	0.00	0.00	194.45	46.74
19	218.40	198.19	231.72	184.85	245.04	177.25	258.33	339.79	271.67	313.00	284.99	137.27
20	135.56	242.74	148.88	102.01	162.19	94.41	175.49	81.16	188.83	67.75	416.13	54.43
21	205.08	184.88	218.40	171.54	231.72	163.93	245.01	313.11	258.35	286.32	271.67	123.96
22	148.90	128.69	162.22	242.72	362.91	107.75	188.83	94.50	202.17	81.09	215.49	67.77
23	191.72	171.51	205.04	158.17	218.35	150.57	231.65	137.32	244.99	123.91	258.31	110.59
24	167.89	147.68	374.37	134.34	401.06	126.74	207.82	113.49	454.35	100.08	481.01	86.76

		To / Bay											
		13	14	15	16	17	18	19	20	21	22	23	24
From / Bay	1	173.97	243.94	94.42	230.67	107.78	211.50	121.12	203.99	134.49	190.60	147.87	171.44
	2	0.00	263.85	114.33	250.58	127.69	231.41	141.03	223.90	154.40	210.51	167.78	191.35
	3	67.77	230.61	174.00	217.34	94.45	198.17	107.79	190.66	121.16	177.27	134.54	158.11
	4	114.41	277.25	127.73	263.98	141.09	244.81	154.43	237.30	167.79	223.91	181.18	204.75
	5	54.41	217.25	67.73	203.98	81.09	184.81	94.43	177.30	107.79	163.91	121.18	144.75
	6	122.15	284.99	135.47	271.72	148.83	252.55	162.17	245.04	175.53	231.65	188.92	212.49
	7	41.09	203.93	54.41	190.66	67.77	171.49	81.11	163.98	94.48	150.59	107.86	131.43
	8	135.47	608.39	309.41	581.84	336.07	543.68	175.49	258.35	188.85	501.75	202.24	225.81
	9	27.75	190.59	41.07	177.32	54.43	158.15	67.77	150.64	81.14	137.25	94.52	118.09
	10	309.52	635.18	162.17	608.64	175.53	279.25	188.87	555.27	202.24	258.35	215.62	490.40
	11	40.68	366.34	27.73	163.98	41.09	144.81	120.72	137.30	67.79	123.91	81.18	221.56
	12	162.15	324.99	175.47	311.72	188.83	292.55	202.17	285.03	215.53	271.65	228.92	517.01
	13	0.00	163.91	14.39	150.64	27.75	131.47	94.04	123.96	54.45	110.57	67.84	91.41
	14	175.53	0.00	188.85	325.10	202.21	305.93	215.56	298.42	469.63	285.03	242.30	265.88
	15	324.99	150.55	0.00	137.27	40.66	118.10	67.36	110.59	41.09	97.21	54.48	78.05
	16	188.81	14.36	202.12	0.00	215.49	319.21	228.83	311.69	242.19	298.31	255.58	570.33
	17	311.69	137.25	325.01	123.98	0.00	104.81	14.43	97.30	27.80	83.91	41.18	64.75
	18	207.79	33.35	221.11	20.08	234.47	0.00	247.82	330.68	261.18	317.29	274.56	298.14
	19	0.00	123.89	311.65	110.62	325.01	91.45	0.00	83.93	14.43	70.55	27.82	51.39
	20	215.49	41.05	228.81	27.77	242.17	8.60	255.51	0.00	268.87	324.99	282.26	305.83
	21	285.01	110.57	298.33	97.30	311.69	78.13	325.03	70.62	0.00	57.23	0.00	38.07
	22	228.83	54.39	242.15	41.11	255.51	21.94	268.85	14.43	282.21	0.00	295.60	319.17
	23	271.65	97.21	284.97	83.93	298.33	64.77	311.67	57.25	325.03	43.87	0.00	24.71
	24	247.82	73.38	261.14	60.10	274.50	40.93	287.84	33.42	301.20	20.04	314.59	0.00

TABLE B-3: Empty travel time between the bays (*sec.*)

From / Bay	To / Bay											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	0.00	318.25	14.49	304.91	27.81	297.31	41.10	284.06	54.44	270.65	67.76	257.33
2	21.08	0.00	34.40	324.82	47.72	317.21	61.01	303.96	74.35	290.56	87.67	277.24
3	325.12	304.92	0.00	291.58	14.48	283.98	27.77	270.73	41.11	257.32	54.43	244.00
4	34.48	14.27	47.80	0.00	61.11	330.61	74.41	317.36	87.75	303.95	101.07	290.64
5	311.76	291.56	325.08	278.21	0.00	270.61	14.41	257.36	27.75	243.95	41.07	230.64
6	0.00	22.01	55.54	8.67	68.85	0.00	82.15	325.10	95.49	311.69	108.81	298.38
7	298.44	278.24	311.76	264.90	325.08	257.29	0.00	244.04	14.43	230.64	27.75	217.32
8	55.53	35.33	68.85	21.99	82.17	14.39	95.47	0.00	108.81	325.01	122.13	311.69
9	285.10	264.90	298.42	251.56	311.74	243.95	325.03	230.70	0.00	217.29	14.41	203.98
10	68.92	48.72	82.24	35.38	95.56	27.77	108.85	14.52	122.19	0.00	135.51	325.08
11	271.76	251.56	285.08	238.22	298.40	230.61	311.69	217.36	325.03	203.95	0.00	190.64
12	82.22	62.01	95.54	48.67	108.85	41.07	122.15	27.82	135.49	14.41	148.81	0.00
13	258.42	238.22	271.74	224.88	285.06	217.27	298.35	204.02	311.69	190.61	325.01	177.30
14	95.60	75.40	108.92	0.00	122.24	54.45	135.53	41.20	148.88	27.80	162.19	14.48
15	245.06	0.00	258.38	211.51	271.69	203.91	284.99	190.66	298.33	177.25	311.65	163.93
16	108.87	88.67	122.19	75.33	135.51	67.73	0.00	54.48	0.00	41.07	0.00	27.75
17	231.76	211.56	245.08	198.22	258.40	190.61	271.69	177.36	285.03	163.95	298.35	150.64
18	127.86	107.66	141.18	94.32	154.50	86.71	167.79	73.46	0.00	0.00	194.45	46.74
19	218.40	198.19	231.72	184.85	245.04	177.25	258.33	164.00	271.67	150.59	284.99	137.27
20	135.56	115.35	148.88	102.01	162.19	94.41	175.49	81.16	188.83	67.75	202.15	54.43
21	205.08	184.88	218.40	171.54	231.72	163.93	245.01	150.68	258.35	137.27	271.67	123.96
22	148.90	128.69	162.22	115.35	175.53	107.75	188.83	94.50	202.17	81.09	215.49	67.77
23	191.72	171.51	205.04	158.17	218.35	150.57	231.65	137.32	244.99	123.91	258.31	110.59
24	167.89	147.68	181.21	134.34	194.52	126.74	207.82	113.49	221.16	100.08	234.48	86.76

		To / Bay											
		13	14	15	16	17	18	19	20	21	22	23	24
From / Bay	1	81.10	243.94	94.42	230.67	107.78	211.50	121.12	203.99	134.49	190.60	147.87	171.44
	2	0.00	263.85	114.33	250.58	127.69	231.41	141.03	223.90	154.40	210.51	167.78	191.35
	3	67.77	230.61	81.09	217.34	94.45	198.17	107.79	190.66	121.16	177.27	134.54	158.11
	4	114.41	277.25	127.73	263.98	141.09	244.81	154.43	237.30	167.79	223.91	181.18	204.75
	5	54.41	217.25	67.73	203.98	81.09	184.81	94.43	177.30	107.79	163.91	121.18	144.75
	6	122.15	284.99	135.47	271.72	148.83	252.55	162.17	245.04	175.53	231.65	188.92	212.49
	7	41.09	203.93	54.41	190.66	67.77	171.49	81.11	163.98	94.48	150.59	107.86	131.43
	8	135.47	298.31	148.78	285.03	162.15	265.87	175.49	258.35	188.85	244.97	202.24	225.81
	9	27.75	190.59	41.07	177.32	54.43	158.15	67.77	150.64	81.14	137.25	94.52	118.09
	10	148.85	311.69	162.17	298.42	175.53	279.25	188.87	271.74	202.24	258.35	215.62	239.19
	11	14.41	177.25	27.73	163.98	41.09	144.81	54.43	137.30	67.79	123.91	81.18	104.75
	12	162.15	324.99	175.47	311.72	188.83	292.55	202.17	285.03	215.53	271.65	228.92	252.49
	13	0.00	163.91	14.39	150.64	27.75	131.47	41.09	123.96	54.45	110.57	67.84	91.41
	14	175.53	0.00	188.85	325.10	202.21	305.93	215.56	298.42	228.92	285.03	242.30	265.88
	15	324.99	150.55	0.00	137.27	14.39	118.10	27.73	110.59	41.09	97.21	54.48	78.05
	16	188.81	14.36	202.12	0.00	215.49	319.21	228.83	311.69	242.19	298.31	255.58	279.15
	17	311.69	137.25	325.01	123.98	0.00	104.81	14.43	97.30	27.80	83.91	41.18	64.75
	18	207.79	33.35	221.11	20.08	234.47	0.00	247.82	330.68	261.18	317.29	274.56	298.14
	19	0.00	123.89	311.65	110.62	325.01	91.45	0.00	83.93	14.43	70.55	27.82	51.39
	20	215.49	41.05	228.81	27.77	242.17	8.60	255.51	0.00	268.87	324.99	282.26	305.83
	21	285.01	110.57	298.33	97.30	311.69	78.13	325.03	70.62	0.00	57.23	0.00	38.07
	22	228.83	54.39	242.15	41.11	255.51	21.94	268.85	14.43	282.21	0.00	295.60	319.17
	23	271.65	97.21	284.97	83.93	298.33	64.77	311.67	57.25	325.03	43.87	0.00	24.71
	24	247.82	73.38	261.14	60.10	274.50	40.93	287.84	33.42	301.20	20.04	314.59	0.00

TABLE B-4: FROM-TO ROUTING PROBABILITIES MATRIX.

Source Bay	Destination Bay — Routing Probability										
1	13 (1)										
2	8 (0.72)	10 (0.14)	11 (0.14)								
3	8 (0.92)	15 (0.08)									
4	8 (0.91)	10 (0.09)									
5	3 (0.25)	10 (0.75)									
6											
7	5 (1)										
8	3 (0.07)	4 (0.11)	9 (0.04)	10 (0.02)	12 (0.02)	14 (0.15)	15 (0.2)	16 (0.11)	17 (0.04)	18 (0.11)	22 (0.13)
9	3 (1)										
10	1 (0.24)	7 (0.12)	11 (0.04)	12 (0.24)	13 (0.12)	14 (0.08)	16 (0.04)	20 (0.08)	24 (0.04)		
11	5 (0.13)										
12	4 (0.09)	8 (0.09)	11 (0.73)	24 (0.09)							
13	10 (0.9)	19 (0.1)									
14	2 (0.1)	11 (0.2)	21 (0.7)								
15	8 (0.5)	17 (0.4)	19 (0.1)								
16	5 (0.33)	24 (0.67)									
17	8 (1)										
18	2 (1)										
19	8 (0.67)	10 (0.33)									
20	2 (0.5)	11 (0.5)									
21	8 (0.86)	10 (0.14)									
22	4 (0.83)	5 (0.17)									
23											
24	3 (0.25)	5 (0.12)	9 (0.25)	11 (0.38)							

REFERENCES

[1] Andradóttir, S. and Bier, V.M. (2000) Applying Bayesian ideas in simulation. Simulation Practice and Theory, 8(3-4): 253-280.

[2] Arzt T., and Bulcke F. (1999). A new low cost approach in 200 mm and 300 mm AMHS. Semiconductor Fabtech, Available online at http://www.fabtech.org.

[3] Aughenbaugh, J. M., and Paredis, C. J. J., (2005), The Value of Using Imprecise Probabilities in Engineering Design, ASME 2005 DETC DTM, Long Beach, CA, USA, pp. DETC 2005-85354.

[4] Batarseh, O.G. and Wang, Y. (2008) Reliable Simulation with Input Uncertainties Using an Interval-Based Approach, In *Proceedings of the 2008 Winter Simulation Conference*, ed. S.J. Mason, R. Hill, L. Moench, and O. Rose.

[5] Batarseh, O.G. and Wang, Y. (2009) Input Uncertainties: An Interval based Simulation and Second Order Monte Carlo Simulation, In *Proceedings of the 2009 Industrial Engineering Research Conference,* eds. J. W. Fowler, and S. J. Mason.

[6] Barton, R. and Schruben, L. (2001) Resampling methods for input modeling. In *Proceedings of the 2001 Winter Simulation Conference,* ed. A.B. Peters, J.S. Smith, D. J. Medeiros, and M. W. Rohrer.

[7] Cardarelli G., and Pelagagge, P. J. (1995). Simulation tool for design and management optimization of automated interbay material handling and storage systems for large wafer fab. IEEE Transactions on Semiconductor Manufacturing, 8, 1:44-49.

[8] Cheng, R. C. H. (1994) Selecting input models. In *Proceedings of the 1994 Winter Simulation Conference,* ed. J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila, 184–191.

[9] Cheng, R.C.H. and Holland, W. (1997) Sensitivity of computer simulation experiments to errors in input data. Statistical Computation and Simulation, 57: 219–241.

[10] Cheng, R.C.H. and Holland, W. (1998) Two-point methods for assessing variability in simulation output. Statistical Computation and Simulation, 60: 183–205.

[11] Cheng, R.C.H. and Holland, W. (2004) Calculation of Confidence Intervals for Simulation Output. ACM Transactions on Modeling and Computer Simulation. Vol. 14, No. 4, 344-362.

[12] Chick, S. E. (1997) Bayesian Analysis for Simulation Input and Output. In *Proceedings of the 1997 Winter Simulation Conference*, S. Andradóttir, K. J. Healy, D. H. Withers and B. L. Nelson (eds.), Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, 253-260.

[13] Chick, S.E. (1999) Steps to implement Bayesian input distribution selection. In *Proceedings of the 1999 Winter Simulation Conference,* ed. Farrington, P.A., Nembhard, H.B., Sturrock, D.T. and G.W. Evans, 317–324.

[14] Chick, S. E. (2000) Bayesian methods for simulation. In *Proceedings of the 2000 Winter Simulation Conference,* ed. J. A. loines, R. R. Barton, K. Kang, and P. A. Fishwick, 109-118.

[15] Chick, S. E. (2001) Input distribution selection for simulation experiments: accounting for input uncertainty. Operations Research, 49: 744-758.

[16] Chick, S.E. (2006) Bayesian Ideas for Discrete Event Simulation: Why, What and How, In *Proceedings of the 2006 Winter Simulation Conference*, ed. R.G. Ingalls, M.D. Rossetti, J.S. Smith, B.A. Peters, 96-106

[17] Cochran, W. G. (1977). Sampling techniques (3rd ed.). New York: John Wiley & Sons.

[18] Deleris, L.A. and Erhun, F. (2005) Risk management in supply networks using Monte-Carlo simulation. In *Proceedings of the 2005 Winter Simulation Conference*, ed. M.E. Kuhl, N.M. Steiger, F.B. Armstrong, and J.A. Joines, 1643-1649

[19] Dempster, A. (1967) Upper and lower probabilities induced by a multi-valued mapping. Annals of Mathematical Statistics, 38(2):325—339.

[20]   Draper, D. (1995) Assessment and propagation of model uncertainty (with discussion). Journal of the Royal Statistical Society, B57(1): 45–97.

[21]   Dubois, D. and Prade, H. (1988) Possibility Theory: An Approach to Computerized Processing of Uncertainty. Plenum, New York.

[22]   Elkins, D., LaFleur, C., Foster, E., Tew, J., Biller, B., and Wilson, J.R. (2007) Clinic: correlated inputs in an automotive paint shop fire risk simulation. In *Proceedings of the 2007 Winter Simulation Conference*, ed. S.G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J.D. Tew, and R.R. Barton. 250-259

[23]   Ferson, S., Kreinovich, V., Hajagos, J., Oberkampf, W., and Ginzburg, L. (2007) Experimental Uncertainty Estimation and Statistics for Data Having Interval Uncertainty. Sandia National Laboratories Technical report SAND2007-0939, Setauket, New York

[24]   Ferson, S., V. Kreinovich, L. Ginzburg, D. S. Myers, and K. Sentz. (2002) Constructing probability boxes and Dempster-Shafer structures. Sandia National Laboratories Technical report SAND2002-4015, Albuquerque, NM.

[25]   Ferson, S. and S. Donald (1998). "Probability Bounds Analysis." In A. Mosleh and R. A. Bari (Eds.), Probabilistic Safety Assessment and Management (pp. 1203-8). New York, NY: Springer-Verlag.

[26]   Ferson S and Ginzburg LR (1996), Different methods are needed to propagate ignorance and variability. Reliability Engineering and System Safety, 54: 133-144.

[27]   Gardenes, E., Sainz, M.A., Jorba, L., Calm, R., Estela, R., Mielgo, H., and Trepat, A. (2001) Modal intervals. Reliable Computing, 7(2): 77-111.

[28]   Glynn, P. (1986) Problems in Bayesian analysis of stochastic simulation. In *Proceedings of the 1986 Winter Simulation Conference,* ed. J. R. Wilson, J. O. Henriksen, and S. D. Roberts, 376-383.

[29]   Golberg, M. A. (1984) An Introduction to Probability Theory with Statistical Applications. . New York and London: Plenum Press.

[30]   Granvilliers, L. Kreinovich, V. and Müller, N. (2003) Novel approaches to numerical software with result verification, In: R. Alt, A. Frommer, R. B.

Kearfott, and W. Luther (eds.), Numerical Software with Result Verification, In Lectures Notes in Computer Science, 2004, Vol. 2991, 274-305.

[31]   Henderson, S. (2003) Input Model Uncertainty: Why do We Care and What Should We Do About It? In *Proceedings of the 2003 Winter Simulation Conference*, ed. S. Chick, P.J. Sanchez, D. Ferrin and D. J. Morrice, 90-100.

[32]   Hsu M., Bhatt M., Adolphs R., Tranel D., and Camerer C., (2005) Neural Systems Responding to Degress of Unertainty in Human Decision-Making. *Science,* **310:** 1680-1683.

[33]   ITRS. (2007). International Technology Roadmap for Semiconductors. Edition: Factory Integration. Available online at http://www.itrs.net

[34]   ITRS. (2005). ITRS 2005 Factory Integration Chapter Material Handling Backup Section. Available online at http://public.itrs.net/reports.html

[35]   Kaucher, E. (1980)   Interval analysis in the extended interval space IR. Computing Supplementa, Vol.2, 33-49

[36]   Klir, G.J. (2006) Uncertainty and Information: Foundations of Generalized Information Theory. New York: Wiley.

[37]   Jones, S. (2003). "300 mm perceptions and realities". Semiconductor International. 26, 69-72.

[38]   Lan, H., Nelson, B.L., and Staum, J. (2007) A confidence interval for tail conditional expectation via two-level simulation. In *Proceedings of the 2007 Winter Simulation Conference*, ed. by S.G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J.D. Tew, and R.R. Barton, 949-957

[39]   Law, A. M. (2007) Simulation Modeling & Analysis. 4th ed. New York: McGraw-Hill.

[40]   Livny, M., Melamed, B., and Tsiolis, A.K. (1993) The impact of autocorrelation on queueing systems. Management Science, 39(3): 322-339

[41]   Lucas, T.W., Sanchez, S.M., Martinez, F., Sickinger, L.R., and Roginski, J.W. (2007) Defense and homeland security applications of multi-agent simulations. In *Proceedings of the 2007 Winter Simulation Conference*, ed. by S.G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J.D. Tew, and R.R. Barton, 138-149

[42]   Molchanov, I. (2005) Theory of Random Sets. London: Springer

[43]   Möller, B. and Beer M. (2004) Fuzzy Randomness: Uncertainty in Civil Engineering and Computational Mechanics. Springer, Berlin.

[44]   Mood A. M., and Graybill F. A. (1963). Introduction to the Theory of Statistics. 2nd Edition Mc-Graw Hill Book Company, Inc.

[45]   Moore, R.E. (1966) Interval Analysis. Prentice-Hall, Englewood Cliffs, N.J.

[46]   National Bureau of Standards (1949). Tables to Facilitate Sequential t-Tests. Washington: US Department of Commerce.

[47]   Neumaier, A. (2004) Clouds, fuzzy sets, and probability intervals. Reliable Computing, 10(4):249—272.

[48]   Ng, S.H. and Chick, S.E. (2006) Reducing Parameter Uncertainty for Stochastic Systems. ACM Transactions on Modeling and Computer Simulation, 16(1): 26-51

[49]   Rushton, S. (1950) On a Sequential t test. Biometrika, 39, 326 -333.

[50]   SIA press release. (2009). Global Semiconductor Sales Fell by 2.8 Percent in 2008.   Available   online   at   http://www.sia-online.org/cs/papers_publications/press_release_detail?pressrelease.id=1534.

[51]   Shafer, G. A. (1990) Mathematical Theory of Evidence, Princeton University Press, Princeton, NJ.

[52]   Stamatelatos M. (2002) Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practictioners. Office of Safety and Mission Assurance NASA Headquarters.

[53]   Stuart, A. and Ord, J. K. (1987) Kendall's Advanced Theory of Statistics. 5th ed. Vol. 1. New York: Oxford University Press.

[54]   Tucker, W.T. & Ferson, S. (2003). Probability Bounds Analysis in Environmental Risk Assessments. Applied Biomathematics, Setauket, NY. Available at: http://www.ramas.com/pbawhite.pdf, accessed on 9 March 2010.

[55]   Vose, D. (2000) Risk Analysis, A Quantitative Guide. 2nd Edition, John Wiley & Sons, NewYork.

[56]   Walley, P. (1991) Statistical Reasoning with Imprecise Probabilities, Chapman & Hall, London.

[57]   Walley, P. (1991) Statistical Reasoning with Imprecise Probabilities, Chapman & Hall, London.

[58]   Wang, Y. (2008a) Imprecise probabilities with a generalized interval form. In R.L. Muhanna and R.L. Mullen, eds., Proc. 3rd Int. Workshop on Reliability Engineering Computing (REC'08), Savannah, Georgia, 45-59.

[59]   Wang, Y. (2008b) Imprecise probabilities based on generalized intervals for system reliability assessment. To appear in International Journal of Reliability & Safety.

[60]   Wang, Y. (2008c) Interpretable Interval Constraint Solvers in Semantic Tolerance Analysis. Computer-Aided Design & Applications, 5(5): 654-666

[61]   Wang, Y. (2008d) Semantic Tolerance Modeling with Generalized Intervals. Journal of Mechanical Design, 130(8): 081701(1-7)

[62]   Weichselberger, K. (2000) The theory of interval-probability as a unifying concept for uncertainty.   International Journal of Approximate Reasoning, 24(2-3): 149-170.

[63]   Xiang, G,. Starks S.A, Kreinovich, V. and Longpre, L. (2007) Computing population variance and entropy under interval uncertainty: linear-time algorithms. Reliable Computing, 13(6): 467-488

[64]   Zimmermann H. J. (2000) An application-oriented view of modeling uncertainty. *European Journal of Operational Research*, **122**: 190-198

[65]   Zouaoui, F., and Wilson, J.R. (2001a) Accounting for input model and parameter uncertainty in simulation. In *Proceedings of the 2001 Winter Simulation Conference*, ed. B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer.

[66]   Zouaoui, F., and Wilson, J.R. (2001b) Accounting for parameter uncertainty in simulation input modeling. In *Proceedings of the 2001 Winter Simulation Conference,* ed. B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer.

[67]   Zouaoui, F. and Wilson, J.R. (2003) Accounting for parameter uncertainty in simulation input modeling. IIE Transactions, 35: 781-792.

[68]   Zouaoui, F. and Wilson, J.R. (2004) Accounting for input-model and input-parameter uncertainties in simulation. IIE Transactions, 36(11): 1135-1151