2013

Interval-based simulation to model input uncertainty in stochastic Lanchester models

Batarseh, Ola

Military Operations Research (A), v.18:no.1, pp.61-76
http://hdl.handle.net/10945/39517

Interval-Based Simulation to Model Input Uncertainty in Stochastic Lanchester Models

Ola Batarseh[1] and Dashi Singham[2]
[1]H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332, ola.batarseh@isye.gatech.edu
[2]Operations Research Department, Naval Postgraduate School, Monterey, CA 93943, dsingham@nps.edu

ABSTRACT

Interval-based simulation (IBS) has been proposed to model input uncertainty in discrete-event simulation. The foundation of this new simulation paradigm is imprecise probability, which models systems under both aleatory and epistemic uncertainties. The statistical distribution parameters in IBS are represented by intervals instead of precise real numbers. This paper discusses how the IBS approach can be applied to stochastic Lanchester models that are used in combat simulation to better account for input parameter uncertainty. The advantages of this approach are explored in comparison with second-order Monte Carlo simulation. Using IBS, an improved estimate of the probability of a team winning a battle is calculated by taking advantage of the interval structure. By resampling from intervals to determine event times, we can separate the effect of parameter uncertainty from random number generator uncertainty to estimate the probability that one team will win for given stream of random numbers used in a single replication. Additionally, we show how our method can be used to improve the reliability of stochastic Lanchester results by accounting for different skill levels within each team, and show how the interval structure can be used to highlight the disproportionate effect of the first few encounters in the battle.

1 INTRODUCTION

Simulation analysts make decisions based on obtained simulation outputs. The reliability of these results depends on the reliability of the input used. The reliability of an input distribution can be assessed by its ability to capture all of the uncertainty components in a simulation. The total uncertainty is composed of two components: *aleatory uncertainty* and *epistemic uncertainty*. Aleatory uncertainty is the variability due to inherent randomness in the system. This component is naturally irreducible and is introduced into the simulation using probability distributions to model input processes. Epistemic uncertainty is due to the lack of perfect information about the system, and is reducible by collecting additional measurements.

The interval-based approach models both uncertainty components explicitly based on the notion of imprecise probability. Imprecise probability assumes that the probability of an event takes a range of values instead of a single value. For example, the probability of an event E is given as $P(E) = [\underline{p}, \overline{p}]$, instead of $P(E) = p$ where $p \in [\underline{p}, \overline{p}]$. The uncertainty in our knowledge is admitted by the interval representation, instead of advocating that a single value should be the only one considered. Batarseh (2010) proposed applying the interval-based approach to discrete event simulation. For this paper, we assume that all intervals considered are proper intervals, where $\underline{p} \leq \overline{p}$. The parameters of the input distributions to the simulation are represented by intervals. For example, in a queueing model, instead of modeling the inter-arrival times as $exp(\lambda)$ (exponentially distributed with rate λ), we would use $exp([\underline{\lambda}, \overline{\lambda}])$. Each inter-arrival time can be represented as a range of values by using the same uniform random number U in $[0, 1]$ to generate $exp(\underline{\lambda})$ and $exp(\overline{\lambda})$. Using the inversion technique to generate exponential random variables results in an inter-arrival time range represented by an interval $[exp(\overline{\lambda}), exp(\underline{\lambda})]$, and this interval is propagated through the simulation. Interval-based simulation (IBS) can help account for the uncertainty in λ.

This paper shows how IBS can be used to relax some of the fixed parameter assumptions employed by discrete-event simulations. Batarseh et al. (2010) shows how IBS can be used in queueing models to better assess the distribution of the mean delay time by incorporating uncertainty in parameters used to generate event times. In this paper, we propose a different use for IBS. We use a series of random numbers to generate the intervals for event times on the future events list, and sample from the ranges of the event times to decide which event is executed first. We can resample from the same intervals to generate new orderings of events. This allows us to separate the effect of the parameter uncertainty (incorporated using interval event times) from the uncertainty due to the random numbers used to drive the simulation event times.

Another way of incorporating input parameter uncertainty is to use second order Monte Carlo (SOMC) methods. This involves sampling the input parameters from a given range or distribution and running the simulation using the sampled parameter values. However, this method still assumes that the input parameters are fixed within a given simulation run. IBS captures the input parameter uncertainty in a different way by varying the parameters within a simulation run. By generating ranges for each event time and sampling event times from the ranges, we allow for different effective parameter values to be used in a given run. The idea of varying parameters within a simulation run is not new. Frequency domain experiments vary the parameters within a single run according to given functions so that meta-models can be constructed from very few runs. For details, see Morrice and Schruben (1993) and Sanchez and Buss (1987). Here, we look at parameter variation within a run to better account for input uncertainty in the real world: not only are we unsure of the real parameter value, but we may not be sure that the parameter remains constant.

Stochastic Lanchester models (SLMs) use probability to model the interactions between members of two opposing teams in a combat situation. Discrete event simulation is one approach that has been used to implement SLMs, and we can enhance traditional discrete-event simulations of SLMs by applying an interval-based approach. Incorporating uncertainty into the parameters of SLMs can be used to improve the reliability of the models. Traditional SLMs assume a fixed value for the skill rate of each competing side. Our proposed approach suggests that the uncertainty in the skill rates of the agents can be modeled using intervals. Here we assume that the skill rates vary between agents and throughout the simulation run. Even if the blue agents are on average more skillful than red agents, a random sampling of a blue and red agent could yield a more skillful red agent defeating a less skillful blue agent in an individual encounter. We can use intervals to capture some of this uncertainty without having to assign each agent a skill rate (as is often done in agent-based simulation).

The interval approach allows us to explicitly model uncertainty in the distribution parameters and to see the effect of a range of parameters *within a single simulation run.* SOMC methods require multiple runs to test different parameters. Additionally, IBS allows for a potentially more realistic model, since constant parameter rates are not assumed. We will show how the interval structure allows us to obtain more information from a single simulation run, since we can resample from the same intervals to obtain different outcomes. For SLMs, we can exploit the interval structure to obtain improved estimates of the probability of a certain side winning. This is done by using a fixed set of random numbers to generate interval event times for individual encounters, and resampling from these intervals to determine the sequence of wins on each side. This allows us to estimate the probability of one side winning based on a fixed set of uniform random numbers used to generate intervals. Without intervals, each independent simulation run of the SLM would return either a 1 (red wins) or a 0 (blue wins). By resampling to generate multiple outcomes, we can estimate the *probability red wins* for that given set of random interval times, providing more information than a simple 1 or 0.

The paper is outlined as follows. Section 2 begins with an overview of IBS. A uniform sampling approach (to sample from intervals for event times) is proposed to advance the simulation clock, though other sampling methods can be used. Section 3 explains the SLM we employ using a discrete-event simulation approach. Section 4 shows how we incorporate IBS into SLMs to obtain an improved estimate for the probability of a certain side winning by exploiting the interval structure of our method. Section 5 shows the experimental results and provides uncertainty quantification for various parameter range choices. Finally, Section 6 concludes the paper and recommends ideas for future work.

2 INTERVAL-BASED SIMULATION (IBS)

2.1 Background

The interval-based simulation mechanism was proposed in Batarseh (2010). The simulation model parameters are based on intervals instead of real numbers in order to obtain more reliable estimates of outputs. This section explains IBS, and gives details about the types of IBS implementations considered in this paper. We compare IBS models and SOMC models.

The two components of total uncertainty, aleatory and epistemic, have been always present in simulation studies. However, epistemic uncertainty is often ignored in simulation practice. Fixing the input parameter values of statistical distributions used in generating simulation input assumes that their values are known with certainty. Basic simulation input modeling techniques that use the maximum likelihood estimator (MLE) reveal a gap in uncertainty quantification for simulation models. The inability of the MLE to represent the epistemic uncertainty in simulation encourages us to find new ways to quantify the total uncertainty. Running simulations with fixed input parameters and relying on the randomness that is caused by the random number generator does not represent the total uncertainty. Imprecise probabilities suggest a solution using intervals (Wang 2008).

Aleatory and epistemic uncertainty can be represented explicitly using IBS and imprecise probability. The stochastic probability distributions for input data represent the aleatory uncertainty. The lack of information on the parameters of the distributions, which is epistemic uncertainty, is captured within the bounds of the intervals on the parameters. To account for both types of uncertainty, probability distributions with interval parameters are used to model input random variables. This representation of the total uncertainty captures both parameter and model uncertainty, since one interval-valued distribution actually models a set of distributions simultaneously.

Uncertainty quantification in simulation has attracted the attention of many researchers. Some of the proposed approaches include SOMC methods (Vose 2008), Bayesian approaches (Glynn 1986, Chick 1997, Zouaoui and Wilson 2001a,b), Delta methods (Cheng and Holland 1997, 1998), and bootstrapping (Barton and Schruben 2001). The SOMC method is a popular way to incorporate epistemic uncertainty using a two-step sampling process. First, a parameter value is sampled from its corresponding distribution, which models the epistemic uncertainty component. Second, this precise value of the parameter is used to run the simulation. Epistemic uncertainty is evaluated by examining a number of replications. In Bayesian approaches, a prior distribution on each input parameter is assigned to describe its initial uncertainty. The prior distribution is then updated to a posterior distribution based on the observed data. In the Delta method, the total simulation output variance is estimated by two terms. The first term is the simulation variance, and the second one is the input parameter variance. In the bootstrap approach, the effect of input parameter uncertainty is quantified. Using available information, the parameters are first estimated by maximum likelihood estimation. The estimates are then used to draw new samples of the observations. Finally, Ingalls et al. (2000) proposed the implementation of temporal intervals in qualitative simulation graphs. The objective of this representation is to improve on the robustness of qualitative simulation by accounting for uncertainty due to the lack of knowledge about the system being modeled. The adopted approach uses intervals as the imprecise specification of event occurrences. The temporal intervals replace the traditional statistical representation of probability distribution functions. The main applications of this approach are in business and project management fields for which qualitative simulation is appropriate.

The interval-based approach was first proposed for the modeling and simulation of queueing systems (for details, see Batarseh et al. (2010)). The two important input parameters of queueing systems are the inter-arrival rates and the service rates. Interval parameters were used for the probability distributions involved in the queueing systems. For instance, exponential distributions of the inter-arrival rate and the service rate were given as $exp([\underline{\lambda}, \overline{\lambda}])$ and $exp([\underline{\mu}, \overline{\mu}])$, respectively, instead of assuming one value for λ and one value for μ. In this paper, we define the interval of times for event I as $[\underline{I}, \overline{I}]$, where the lower and upper values for the interval are calculated by using the same uniform random number to generate $\underline{I} = exp(\overline{\lambda})$ and $\overline{I} = exp(\underline{\lambda})$. This representation provides a lower and upper bound for the input distributions as shown in the cumulative distribution functions (cdfs) of Figure 1. In the literature, the lower and the upper bounds of cdfs are sometimes called a p-box (Ferson and Donald 1998). P-boxes use arithmetic and logical calculations in risk analysis or in quantitative uncertainty modeling where numerical calculations must be performed.

Figure 1 can be read in two equivalent ways. At a certain cdf value, the value of the random variate is given as an interval $[\underline{I}, \overline{I}]$. Also, the cdf of a random variate I is given imprecisely as $[\underline{F}, \overline{F}]$. The uncertainty in the input distribution parameters is used to determine ranges for event times and these ranges are propagated through the simulation. From the IBS point of view,

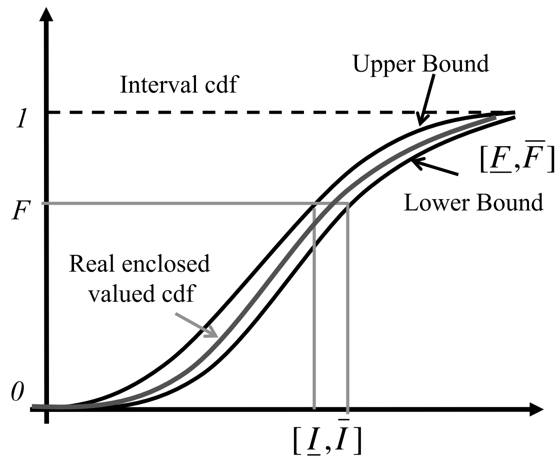

Figure 1: Lower and upper bound cumulative distribution functions for an interval parameter representation.

interval-valued random variates should be generated because real-valued ones are often insufficient to quantify uncertainty. In the example of queueing systems, interval inter-arrival times $[\underline{ia}_i, \overline{ia}_i]$ are generated from their corresponding interval distributions. Given an exponential distribution for inter-arrival times with rates $[\underline{\lambda}, \overline{\lambda}]$, the random interval inter-arrival variates can be generated using the inverse transform method, i.e. $[\underline{ia}_i, \overline{ia}_i] = [-(\ln(1-u_i))/\overline{\lambda}, -(\ln(1-u_i))/\underline{\lambda}]$, where u_i is sampled from a standard uniform distribution. Interval random variates for service times can be generated similarly. Each generated event in the simulation has an associated interval for its possible time of occurrence. Here, exponential distributions are assumed for simplicity, but any distribution function can be used. The use of intervals to manage event times in discrete-event simulation requires new approaches to advance the simulation clock. The following section describes the ways in which the clock can be advanced when IBS is used.

2.2 Sampling Approach to Advance the Simulation Clock

The simulation clock is a variable that gives the current value of the simulation time. In discrete-event simulation, the clock is updated based on the next-event-time advancement approach. When uncertainty is incorporated in the next-event-time, there is no one specific time attributed to each event. This increases the reliability of the simulation and the complexity in deciding which event will occur first. The partial order between the intervals allows for different rules to be used to determine the event order. For example, the values of the lower bound or the upper bound for event times can determine which event occurs first. Figure 2 displays the six different ways two arbitrary intervals $[\underline{I}_1, \overline{I}_1]$ and $[\underline{I}_2, \overline{I}_2]$ for events I_1 and I_2 can be arranged.

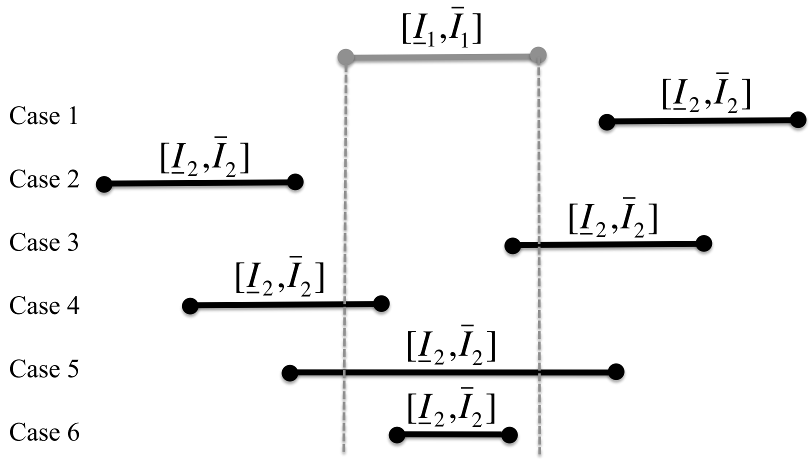

Figure 2: Six possible arrangements for two interval event times.

For each case, we ask the question: should we execute event I_1 or I_2 first? For case 1, it seems clear that we would execute I_1 first, and for case 2 we would execute I_2 first. This question becomes more interesting when the overlapping cases 3,4,5, or 6 occur. A second question is: once we decide which event to execute first, at what time would we execute each event? Three proposed approaches have been investigated to determine the sequence of events in IBS (Batarseh 2010). One possible way is to use the lower bounds of the event times; a second approach is to use the upper bounds of the event times. The third method is to uniformly sample from each interval and execute the event with the smallest sampled time first. In this paper, we adopt the uniform sampling approach to advance the simulation clock.

The uniform sampling approach dictates that each event execution time will be uniformly sampled from its corresponding interval. The events are modeled to happen in interval times to account for parameter uncertainty and each value in the interval is equally likely to be chosen as the time of event execution, so the time for execution of event I_1 is distributed as $s_1 \sim U(\underline{I}_1, \overline{I}_1)$. We note that other distributions can be used as well, though we employ the uniform distribution in this paper. The partial order relation between real-valued numbers sampled from different intervals is straightforward. If the sampled observations occur as $s_1 < s_2$, where $s_2 \sim U(\underline{I}_2, \overline{I}_2)$, then the intervals are ordered as $[\underline{I}_1, \overline{I}_1]$ before $[\underline{I}_2, \overline{I}_2]$ and event I_1 is executed first at time s_1, followed by I_2 at time s_2 (assuming no other events are scheduled). This sampling approach can result in different sequences of events if we were to resample from the same intervals. This is a unique advantage of the IBS implementation proposed in this paper, and is why we maintain the interval structure.

SOMC methods use two stages of sampling to incorporate different possible parameter values. In the first stage, parameter values are sampled. In the second stage, one replication of the simulation is run using the sampled parameters. The uncertainty in the parameter values is quantified across the different simulation runs that use different parameter values. In IBS, the parameter uncertainty is incorporated in a single run by sampling from the interval that represents a range of potential distributions. Using the same set of random numbers to generate the event time intervals, different resamplings from these intervals yield different event execution times and event orderings. The different outcomes that are possible using resamplings are representative of the parameter uncertainty in the model, and we isolate this uncertainty from the random numbers used to generate the intervals. Replications in SOMC experiments require different random numbers to generate different outcomes for different simulation runs. This accounts for the variability due to the choice of probability distribution, but does not account for the parameter uncertainty within a run. IBS and SOMC are compared with respect to their uncertainty quantification in Figure 3.

3 STOCHASTIC LANCHESTER MODELS

Lanchester equations were introduced to model combat based on the initial force sizes and skill rates of two sides engaging in battle. Stochastic lanchester models (SLMs) were introduced to

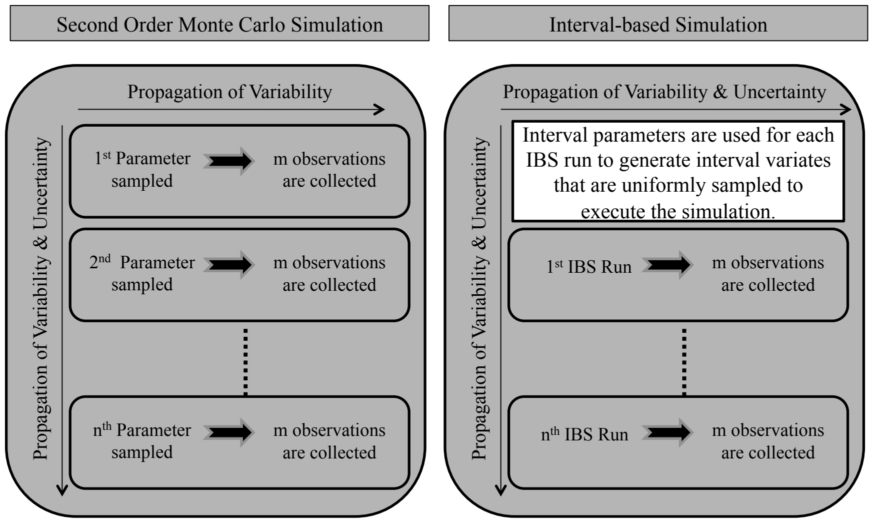

Figure 3: Comparison of SOMC and IBS methods. IBS incorporates input uncertainty within a simulation run, while SOMC only accounts for it across runs.

produce estimates of the probability that a certain side will win the battle. Instead of a deterministic model in which only one outcome is constructed, SLMs allow for different outcomes in each simulation replication, resulting in a distribution of outcomes. In this paper, we use the basic SLM that employs a Markovian assumption. Each side starts with given force sizes, R_0 and B_0, and with skill rates for the agents, λ_R and λ_B. The time until the next death occurs, given current force sizes R and B, is assumed to be exponentially distributed with rate $\lambda_R R+\lambda_B B$. The probability that the next death is suffered by the red team is $\lambda_B B/(\lambda_R R + \lambda_B B)$, with the corresponding probability that blue suffers the death as $\lambda_R R/(\lambda_R R + \lambda_B B)$. Equivalently, this process can be constructed using competing exponentials where the two sides compete with rates $\lambda_R R$ and $\lambda_B B$. The simulation progresses by simulating these individual encounters, and decrementing the force size for the team that suffers a loss, thus reducing that team's likelihood of winning the next encounter.

Stochastic combat models are often used because they provide probability distributions for potential outcomes, while deterministic models are often too simple to provide real predictive value (Lucas 2000). There are a variety of models that attempt to model combat in a deterministic or stochastic manner. Koopman (1970) modeled combat as a set of states and derived transition probabilities. Agent-based models have also become popular as a way of studying particular aspects of combat models (Lauren 2002) since they allow different agents to directly react to one another, rather than assuming aggregate behavior over a time period. Here, we consider the exponential stochastic Lanchester model (ESL), described in the previous paragraph, to show a simple application of IBS. Much combat modeling research involves testing various scenarios using the ESL model, since it is fundamental and easily understood (see, for example, McNaught (1999)). However, the ESL model makes many assumptions that are unrealistic and much work remains to develop better combat models (Ancker Jr. 1995). The point of this paper is to show how IBS can be used to improve simulation models, not to promote use of a specific type of combat model.

Kingman (2002) also analyzes a Markov model for combat, and notes the importance of using stochastic models. Even if the two teams start out evenly matched, the result of the first few encounters can give one side a big advantage for the rest of the battle. We will see in our results that even if both sides are equally matched and have equal probabilities of winning across independent replications, each individual simulation run can be heavily skewed towards one side based on the first few events. One way SLMs are studied is by varying the parameters to estimate the probability that one side (say, red) wins under various conditions (for an example, see Kress and Talmor (1999)). We will use IBS in order to assess the affect of changing the parameters within a simulation run on the probability that a certain side wins.

Figure 4 shows the event graph for the SLM before IBS is applied (for details on event graphs, see Schruben (1983)). Encounters between red and blue agents are generated until one side is left with zero agents. The winner of each encounter is determined using competing exponentials. The team with a larger force size and higher skill rate has a greater chance of generating the smaller exponential value. By repeating this experiment, the probability of one side winning the battle can be estimated by recording the proportion of times red wins. The next section shows how we can apply IBS to these types of combat models.

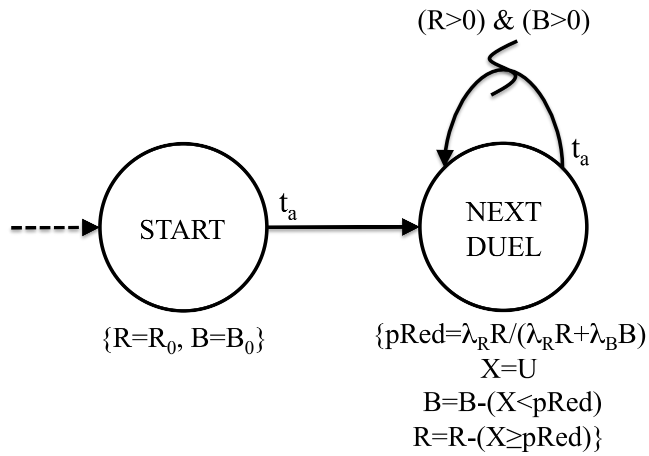

Figure 4: Event relationship graph for the SLM. The time delay t_a is an exponential random variable with rate $\lambda_R R + \lambda_B B$, and $U \sim Unif(0, 1)$.

4 STOCHASTIC LANCHESTER MODELS USING INTERVAL-BASED SIMULATION

In this section, we show how IBS can be used to improve SLMs to obtain estimates of the probability of winning. We start by describing a SLM that incorporates uncertainty in the skill of an agent using intervals, and show how this can affect the probability that an agent wins a particular encounter. We then show how the interval structure of the simulation can be exploited to get improved estimates of the probability of one side winning.

4.1 Model Description

IBS can be applied to SLMs by establishing ranges for the skill parameters. For example, instead of fixing the skill rate of the blue agents (λ_B) at 0.25, we allow the skill rate to vary between $[0.20, 0.30]$ for the different encounters. This allows for a more realistic model, in which all agents do not have the same skill rate throughout the simulation. A SOMC experiment would require that different experiments to be run for different values of λ_B and λ_R, and each experiment would assume that these parameters were fixed throughout.

Suppose we have a range of skill rates to consider for each side, so we use $[\underline{\lambda}_R, \overline{\lambda}_R]$ for λ_R, and

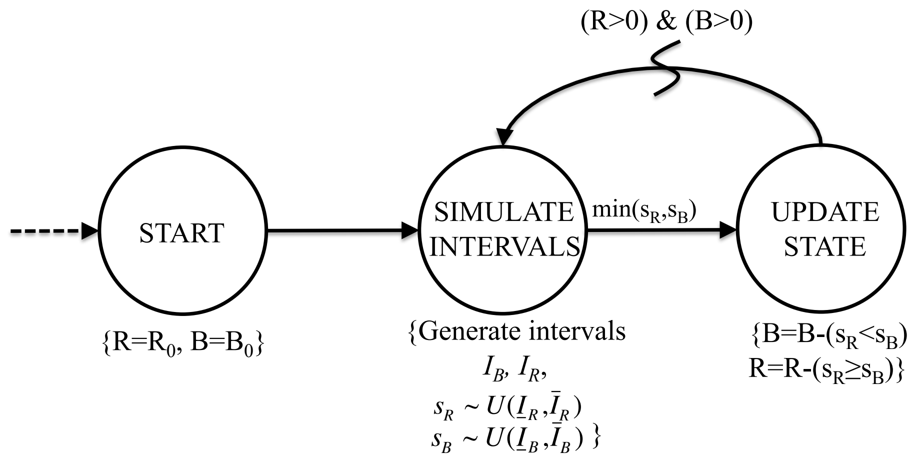

Figure 5: Event relationship graph for SLMs using IBS.

$[\underline{\lambda}_B, \overline{\lambda}_B]$ for λ_B. When simulating a potential encounter between a red and blue agent, we start by simulating a standard uniform random number for each side, U_R and U_B. Then, assuming exponentially distributed inter-arrival times, we can calculate intervals that "compete" to determine which side succeeds. Using our notation, and current force sizes of R and B, the intervals for the next agent attack times for red (I_R) and blue (I_B) are:

$$\left[\frac{-\ln(1-U_R)}{\overline{\lambda}_R R}, \frac{-\ln(1-U_R)}{\underline{\lambda}_R R}\right], \quad \text{and} \quad \left[\frac{-\ln(1-U_B)}{\overline{\lambda}_B B}, \frac{-\ln(1-U_B)}{\underline{\lambda}_B B}\right]. \tag{1}$$

We name the lower bound of I_R as $\underline{I}_R$ and the upper bound as $\overline{I}_R$. Similarly, we can write the interval I_B as $[\underline{I}_B, \overline{I}_B]$. In order to determine which side wins the encounter, we sample uniformly from each interval. Call these samples s_R and s_B, where $s_R \sim U(\underline{I}_R, \overline{I}_R)$ and $s_B \sim U(\underline{I}_B, \overline{I}_B)$. If $s_R < s_B$, then red wins, and blue wins otherwise. Once the force sizes have been updated, both intervals are removed from the events list and the process is repeated with the new force sizes. New intervals for blue and red are generated using the updated force levels, and the process is repeated until one side has reached its termination criteria (loss of all agents). When the termination criteria is achieved, the winning side is recorded for that simulation run. The event graph for the SLM that uses intervals is shown in Figure 5.

4.2 Probability of Winning an Encounter

As described in the previous section, in order to determine which side wins in a particular encounter, we sample uniformly from the intervals $[\underline{I}_R, \overline{I}_R]$ and $[\underline{I}_B, \overline{I}_B]$ and the side with the smallest time succeeds. The side with the larger force size and higher skill level is more likely to generate an interval containing values closer to zero, so it is more likely to win. For given intervals I_R and I_B, we can analytically determine the probability that red wins the encounter using mathematics from IBS. There are six possible arrangements for the intervals as shown in Figure 2. Table 1 shows the probability of red winning the encounter for each of the given cases, where I_1 is I_R and I_2 is I_B. The derivations of these values, taken from Batarseh (2010), are provided in Appendix 1.

Table 1: Probability of red winning the encounter for each interval case.

Case	$P(s_R < s_B)$
1	1
2	0
3	$1 - 1/2 \times (\overline{I}_R - \underline{I}_B)/(\overline{I}_R - \underline{I}_R) \times (\overline{I}_R - \underline{I}_B)/(\overline{I}_B - \underline{I}_B)$
4	$1/2 \times (\overline{I}_B - \underline{I}_R)/(\overline{I}_R - \underline{I}_R) \times (\overline{I}_B - \underline{I}_R)/(\overline{I}_B - \underline{I}_B)$
5	$1/2 \times (\overline{I}_R - \underline{I}_R)/(\overline{I}_B - \underline{I}_B) + (\overline{I}_B - \overline{I}_R)/(\overline{I}_B - \underline{I}_B)$
6	$1/2 \times (\overline{I}_B - \underline{I}_B)/(\overline{I}_R - \underline{I}_R) + (\overline{I}_B - \overline{I}_R)/(\overline{I}_R - \underline{I}_R)$

For each step in the simulation run, Table 1 shows the probability that red will win for a given interval arrangement. In the case of SLMs, the use of these intervals can have real meaning. Agents may have different skill levels, and the sampling from intervals allows us to directly compare two agents of different skill levels from either side. The probabilities in Table 1 accounting for these variations give the weaker side a chance to win an encounter if the intervals overlap.

4.3 Uncertainty Quantification in a Simulated Battle

In most SLM experiments, each independent run produces an outcome of red winning or losing. If the same stream of random numbers were used to generate the random variables that determine encounter times and the probability of red winning each encounter, the same sequence of events could be replicated. However, now instead of single numbers representing the times of the next event, we are sampling from competing intervals determining which side wins each encounter. These intervals are the result of the uncertainty in the skill level of the particular individuals involved. As Table 1 shows, for each simulated encounter, red will win with a certain probability depending on the positioning of the two intervals. The outcome depends on the sample drawn from each interval, the winner is no longer simply determined by the smaller of two simulated exponential random variables, as in the basic discrete-event SLM.

Using these intervals, we can observe different outcomes for the battle using the same stream of random numbers to generate the intervals $[\underline{I}_R, \overline{I}_R]$ and $[\underline{I}_B, \overline{I}_B]$ throughout the experiment. We use the same values of U_R and U_B to calculate (1), but different uniform random numbers to generate s_R and s_B to sample from the intervals. Different samplings of s_R and s_B will lead to different winners in individual encounters, leading to different values of R and B that are used in future calculations of $[\underline{I}_R, \overline{I}_R]$ and $[\underline{I}_B, \overline{I}_B]$. For a given series of random numbers U_R and U_B used to calculate $[\underline{I}_R, \overline{I}_R]$ and $[\underline{I}_B, \overline{I}_B]$, we can simulate a variety of potential outcomes using different samplings from the intervals. Figure 6 shows how different events can be generated from resampling from the same set of overlapping intervals.

Thus, we can estimate the probability that red wins given a particular stream of random uniform numbers U_R and U_B by repeating the simulation many times using these random numbers to generate the intervals, but using new random numbers to sample from the intervals. Averaging over these resamplings returns the probability red wins for the given stream of random uniform numbers U_R and U_B. Instead of delivering a 0 or 1 at the end of each simulation run (1 meaning red wins, 0 meaning red loses), we deliver a probability that red won. The probability is the result of the uncertainty in the skill of the agents that affects the sampling from the intervals. If one simulation run delivers the probability of red winning as 0.9, then given the uncertainty due to the overlap of the intervals in that run, red would win with probability 0.9.

The advantages of delivering the interval probability of red winning include the fact that we obtain

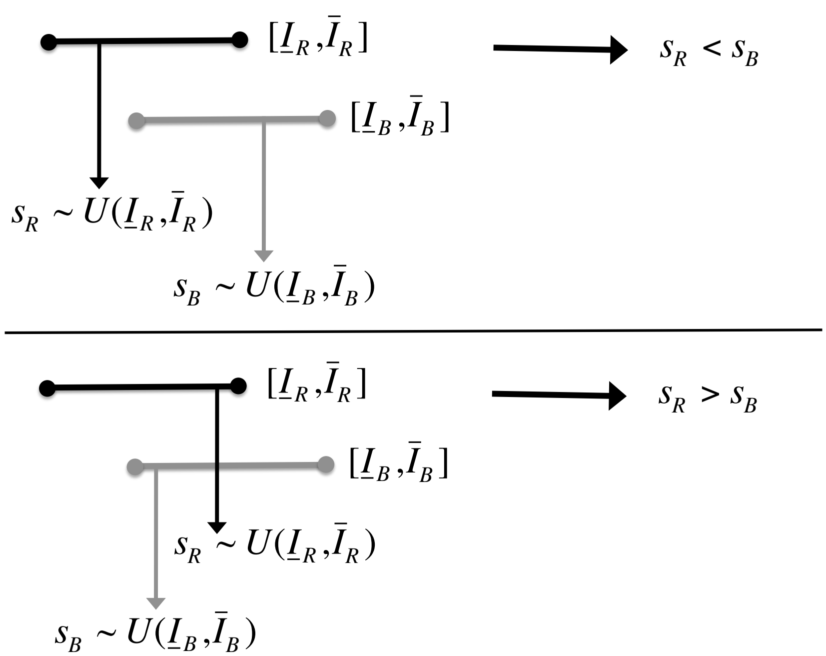

Figure 6: Different samplings lead to different events occurring due to overlapping intervals.

more information from one simulation run (though admittedly at a higher computational cost). If the probability that red won for that simulation run is 0.9, we know not just whether red won or lost for a particular set of interval samplings, but that red will win most of the time given the uncertainty in the skill rate parameters. The entire experiment can be repeated with new values of U_R and U_B to generate a new probability of red winning. Instead of collecting a sequence of zeros and ones, we collect a sequence of probabilities. Averaging over these probabilities for the independent runs gives an estimate of the probability that red will win for given starting values of R_0, B_0, λ_R and λ_B. Our confidence intervals for the true probability of red winning will be narrower than one calculated from a sequence of zeros and ones, since we are delivering better estimates for each replication. The distribution of these probabilities also gives us additional information on how often red is likely to succeed, while taking into account the uncertainty in red and blue's skill rates.

Resampling in this way also allows us to observe the disproportionate effect of the first few encounters on the rest of the battle, as observed in Kingman (2002). If we use the same random numbers to generate a sequence of intervals for the red and blue team, the first intervals will always be the same since they are based on the initial conditions. If red happens to win the first encounter, it will have an increased chance of winning the next one. As one side builds up momentum, it will likely continue to win as the force level of the opposing side declines. By resampling from the first interval (and then from the remaining intervals), we allow different outcomes to happen as either side may obtain the early advantage. Taking the probability that one side wins after aggregating over all the resamplings allows us to see how often victory would occur given those random numbers. In the same way that simulation experiments are replicated to avoid making a decision based on one outcome that could be irregular, here we resample over a given run to get a better idea of how often one side would win given the parameter uncertainty.

5 RESULTS AND IMPLICATIONS FOR STOCHASTIC LANCHESTER MODELS

This section compares IBS and SOMC for SLM simulations. For both methods, we use intervals to represent the range of skill levels among the agents. IBS uses the intervals to advance the simulation in the way described in Section 4, while SOMC samples a skill rate from the intervals at the beginning of each replication and keeps those skill rates constant throughout the run. For each experiment, let p_R be the probability that red wins the battle, and p_B be $1 - p_R$.

In our simulation experiment, we consider three different scenarios. The first scenario, which we call *Equal Teams*, sets the initial force sizes and skill rate intervals to be equal at the start of the experiment, so $R_0 = B_0$ and $[\underline{\lambda}_R, \overline{\lambda}_R]$=$[\underline{\lambda}_B, \overline{\lambda}_B]$. We know that p_R is 0.5 in this case. For the second scenario, we choose starting parameters according to parity points that have been established for deterministic Lanchester models under Lanchester's Square Law. This law states that if red has N times the initial force size of blue, then blue needs a skill rate N^2 times red's skill rate in order for the deterministic model to predict a draw. We call this scenario the *Parity Point* scenario and use a value of $N = 3$, as is often used in the literature (Kress and Talmor 1999, McNaught 1999). The deterministic results do not translate directly to SLMs, so we cannot say anything about what p_R should be, though we anticipate it being around 0.5. The final scenario is *Red Advantage*, in which both sides have equal skill rate intervals, but red starts with a larger force size. We expect p_R to be greater than 0.5 in this case. The parameters chosen are summarized in Table 2.

Table 2: Parameters used for three scenarios.

	Equal Teams	Parity Point	Red Advantage
B_0	30	10	25
R_0	30	30	30
$[\underline{\lambda}_B, \overline{\lambda}_B]$	[0.2,0.3]	[1.8,2.7]	[0.2,0.3]
$[\underline{\lambda}_R, \overline{\lambda}_R]$	[0.2,0.3]	[0.2,0.3]	[0.2,0.3]

5.1 Comparing IBS and SOMC

For the IBS experiments, we ran the SLM 10,000 times for each scenario using different random numbers to generate the intervals. For each of the 10,000 independent replications, we resampled from the intervals 1,000 times to generate an estimate of red winning for that replication (call these values $p_R(i)$, for $i = 1, \ldots, 10,000$). For the SOMC experiments, we sampled the skill rates from $[\underline{\lambda}_B, \overline{\lambda}_B]$ and $[\underline{\lambda}_R, \overline{\lambda}_R]$ 10,000 times. For each sampling, we ran the SLM 1,000 times using the sampled parameters as constants. In this way, we could calculate the probability of red winning for each sampling as $p_R(i)$ for $i = 1, \ldots, 10,000$. By aggregating over these values, we could estimate p_R for each method and for each scenario. The mean and standard deviations of p_R for each scenario and method are presented in Table 3, along with the means and standard deviations of the number of encounters required to end the battle (call this $T_{end}(i)$). For each replication i, $T_{end}(i)$ is found by taking the mean of the 1,000 replications in the inner loop of each method.

Table 3: Results from SLM experiments for the three scenarios.

	IBS - with resampling			SOMC Results		
	Equal	Parity	Red Adv.	Equal	Parity	Red Adv.
Mean $p_R(i)$	0.501	0.477	0.881	0.502	0.472	0.850
St. Dev. $p_R(i)$	0.428	0.437	0.271	0.199	0.150	0.114
Mean T_{end}	46.876	30.088	38.931	46.081	29.843	38.798
St. Dev. T_{end}	4.548	5.775	5.360	0.789	1.779	21.333

From Table 3, we see that the mean value of p_R for the *Equal Teams* scenario is approximately 0.5, for *Parity Point* is around 0.47, and for *Red Advantage* it is much greater than 0.5 (between 0.8 and 0.9). Both IBS and SOMC deliver similar mean values for p_R and T_{end}. This is likely because even though both methods incorporate input uncertainty differently, with enough replications, both will converge to the results for the average skill rate of the agents. We see a major difference in the standard deviation values; the IBS results have a much higher variance. Figure 7 shows the density estimation plots of the $p_R(i)$ values collected for each experiment, which helps explains why the standard deviation is higher for the IBS experiments. These plots were calculated using the density estimation function in the R software program.

We would expect the shape of the density plots in Figure 7 to be different for the IBS and SOMC methods, since they are fundamentally different experiments. For the IBS method, each $p_R(i)$ is formed by averaging over resampling of intervals, using the same random numbers to generate the intervals. The SOMC found each $p_R(i)$ by averaging over independent replications of the SLM using the same skill parameters for each sub-experiment. The SOMC plots show the distribution of $p_R(i)$ given that the skill rates are fixed for the battle, and we do not know what their true values will be. The IBS plots show the distribution given that we are uncertain about the true skill rates and want to incorporate that uncertainty throughout the battle. The U-shaped density curve suggests that for most replications, the battle is one-sided given the uncertainty in the parameters, *even when both teams start out with equal resources*. This means that even if both teams start out with the exact same chance of winning, the sampling based on the first few intervals will greatly affect the rest of the battle if one side happens to win the first few individual encounters.

For the *Equal Teams* scenario, the overall probability of red winning is 0.5 across independent replications, but for a given battle, the result can be very one sided if we are unsure about the skill level. Higher uncertainty means a greater chance of overlap in the intervals. This uncertainty in the parameters results in higher standard deviations for the values of p_R and the length of the simulation (T_{end}). In modeling combat, we are usually interested in what can happen in a single battle, not in averaging over many independent battles. Thus, even if the teams have equal resources at the start

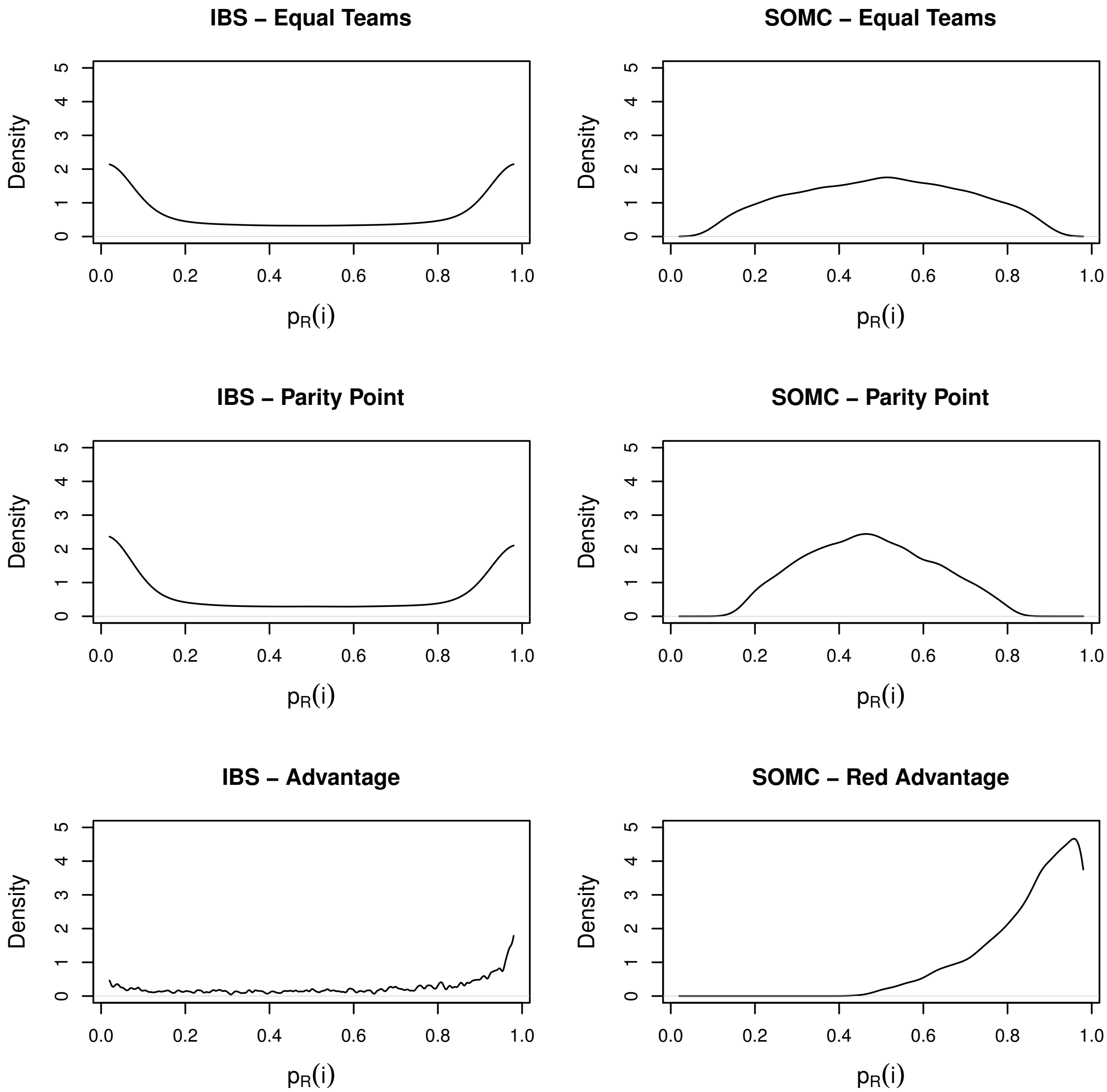

Figure 7: Density estimation plots for $p_R(i)$ for each experiment.

of the battle, Figure 7 shows us that we should not simply take the probability of winning as 0.5, but should realize that the first few events will likely make the result very one sided if there is any input uncertainty in the model. While the IBS method does not provide a narrower estimate of the probability of winning, we believe it provides a more realistic one because it incorporates input uncertainty.

5.2 Effects of resampling

In the previous section, we show that IBS provides a more realistic representation of the variance in the output by incorporating input uncertainty. However, resampling over the intervals actually provides us with less variance than using IBS without resampling. If we were only to run an IBS replication once and collect a 0 or 1 depending on who wins, we have less information than if we resample to get a better estimate of $p_R(i)$. The results from IBS with resampling are reproduced below along with results without resampling. We see that the standard deviations are lower with resampling because have have better estimates for the probability of red winning at each replication. While IBS provides a more realistic estimate (compared to SOMC) with higher variance due to input uncertainty, resampling can be used to mitigate some of that variance.

Table 4: Results from including resampling in IBS models.

	IBS - with resampling			IBS - no resampling		
	Equal	Parity	Red Adv.	Equal	Parity	Red Adv.
Mean $p_R(i)$	0.501	0.477	0.881	0.495	0.476	0.879
St. Dev. $p_R(i)$	0.428	0.437	0.271	0.500	0.499	0.327
Mean T_{end}	46.876	30.088	38.931	46.8091	30.040	38.969
St. Dev. T_{end}	4.548	5.775	5.360	5.396	6.528	5.821

6 CONCLUSIONS AND FUTURE WORK

IBS can be used to incorporate input uncertainty into discrete-event simulation models. In this paper, we used intervals to model uncertainty in the skill level of agents on two teams in a SLM model. This allowed us to account for input uncertainty throughout the simulation run, rather than using fixed skill rates. Additionally, we were able to exploit the interval structure of the simulation model to resample from the intervals to obtain different sequences of events. This allowed us to see multiple battle outcomes for a given stream of random numbers used to calculate event intervals, and the difference in the outcomes is the result of input uncertainty. The results from the IBS experiments showed more variation than those using the standard SOMC method, suggesting that we can quantify the increased uncertainty associated with varying the skill rates throughout a simulation run.

For SLMs, this type of simulation showed how battles involving equally equipped teams can become one sided because one side happens to win the first few encounters, thus increasing their chances of winning the next ones. Increased input uncertainty increases the impact of this effect. Since a skill rate may be hard to quantify, we hope that practitioners will consider using intervals in their model as a simple way of representing this uncertainty. Many military models could benefit from using intervals to represent input uncertainty, creating more reliable models. Future work would extendt the notion of resampling from intervals to other models (say, queueing models or more advanced combat models). The theory behind the proposed resampling method could be developed to make it generally applicable to all models.

References

Ancker Jr., CJ. 1995. A proposed foundation for a theory of combat. *Naval Research Logistics (NRL)* **42** 311–343.

Barton, R.R., L.W. Schruben. 2001. Resampling methods for input modeling. *Proceedings of the 33nd conference on Winter simulation*. IEEE Computer Society, 372–378.

Batarseh, O.G. 2010. Interval-based approach to model input uncertainties in discrete event simulation. Ph.D. thesis, University of Central Florida.

Batarseh, O.G., D. Nazzal, Y. Wang. 2010. An interval-based metamodeling approach to simulate material handling in semiconductor wafer fabs. *Semiconductor Manufacturing, IEEE Transactions on* **23** 527–537.

Cheng, R.C.H., W. Holland. 1997. Sensitivity of computer simulation experiments to errors in input data. *Journal of Statistical Computation and Simulation* **57** 219–241.

Cheng, R.C.H., W. Holland. 1998. Two-point methods for assessing variability in simulation output. *Journal of Statistical Computation and Simulation* **60** 183–205.

Chick, S.E. 1997. Bayesian analysis for simulation input and output. *Proceedings of the 29th conference on Winter simulation*. IEEE Computer Society, 253–260.

Ferson, S., S. Donald. 1998. Probability bounds analysis. A. Mosleh, R.A. Bari, eds., *International Conference on Probabilisitic Safety Assessment and Management.*. Springer Verlag, New York, NY, 1203–1208.

Glynn, P.W. 1986. Problems in bayesian analysis of stochastic simulation. *Proceedings of the 18th conference on Winter simulation*. ACM, 376–379.

Golberg, M.A. 1984. *An introduction to probability theory with statistical applications*. Plenum Press.

Ingalls, R.G., D.J. Morrice, A.B. Whinston. 2000. The implementation of temporal intervals in qualitative simulation graphs. *ACM Transactions on Modeling and Computer Simulation (TOMACS)* **10** 215–240.

Kingman, JFC. 2002. Stochastic aspects of Lanchester's theory of warfare. *Journal of applied probability* **39** 455–465.

Koopman, B.O. 1970. A study of the logical basis of combat simulation. *Operations Research* **18** 855–882.

Kress, M., I. Talmor. 1999. A new look at the 3: 1 rule of combat through Markov stochastic Lanchester models. *Journal of the Operational Research Society* **50** 733–744.

Lauren, MK. 2002. Firepower concentration in cellular automaton combat models-an alternative to Lanchester. *The Journal of the Operational Research Society* **53** 672–679.

Lucas, T.W. 2000. The stochastic versus deterministic argument for combat simulations: Tales of when the average won't do. *Military Operations Research* **5** 9–28.

McNaught, KR. 1999. The effects of splitting exponential stochastic Lanchester battles. *The Journal of the Operational Research Society* **50** 244–254.

Morrice, D.J., L.W. Schruben. 1993. Simulation factor screening using harmonic analysis. *Management Science* **39** 1459–1476.

Sanchez, P.J., A.H. Buss. 1987. A model for frequency domain experiments. *Proceedings of the 19th conference on Winter simulation*. ACM, 424–427.

Schruben, L. 1983. Simulation modeling with event graphs. *Commun. ACM* **26** 957–963.

Vose, D. 2008. *Risk analysis: a quantitative guide*. John Wiley & Sons Inc.

Wang, Y. 2008. Imprecise probabilities with a generalized interval form. *Proc. 3rd Int. Workshop on Reliability Engineering Computing (REC'08), Savannah, Georgia*. 45–59.

Zouaoui, F., J.R. Wilson. 2001a. Accounting for input model and parameter uncertainty in simulation. *Proceedings of the 33nd conference on Winter simulation*. IEEE Computer Society, 290–299.

Zouaoui, F., J.R. Wilson. 2001b. Accounting for parameter uncertainty in simulation input modeling. *Proceedings of the 2001 Winter Simulation Conference*, vol. 1. IEEE, 354–363.

APPENDIX 1

The probability of advancing interval $[\underline{I}_1, \overline{I}_1]$ over $[\underline{I}_2, \overline{I}_2]$ for events I_1 and I_2 for different interval positions is given in Table 1. A ratio distribution (Golberg 1984) is applied here to find the desired probability as explained above. This distribution is constructed from the ratios of two uniformly distributed random variables. Here, we derive the probability of advancing event X ahead of event Y (instead of I_1 and I_2 for notational simplicity). The density functions of the event 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 density functions, which have the same parameters where the minimum value is l and the maximum is k and $l, k \geq 0$. 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 $U = T_x/T_y$ and has a probability distribution function of

$$f_u(u) = \left\{ \begin{array}{ll} \frac{1}{2(k-l)^2}\left(k^2 - \frac{l^2}{u^2}\right), & \frac{1}{k} \leq u \leq 1 \\ \frac{1}{2(k-l)^2}\left(\frac{k^2}{l^2} - l^2\right), & 1 \leq u \leq \frac{k}{l} \end{array} \right\}. \tag{2}$$

The derivation of (2) 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:

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

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

$$P\{U \leq u\} = \int_{G_1} f_{T_x}(t_x) f_{T_y}(t_y) dt_x dt_y + \int_{G_2} f_{T_x}(t_x) f_{T_y}(t_y) dt_x dt_y. \tag{3}$$

We evaluate (3) for $(l/k \leq u \leq 1)$ and for $(1 \leq u \leq k/l)$. The regions G_1 and G_2 are shown in Figure 8. First, we show the evaluation of the first region, $(l/k \leq u < 1)$. The double integration

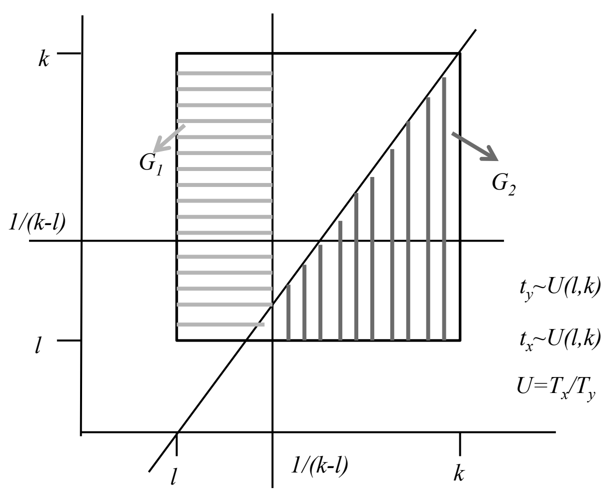

Figure 8: Integration region for calculating the distribution of Tx/Ty.

gives

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

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

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

We evaluate over the region $(1 \leq u \leq k/l)$ to get

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

The derivative of the above cumulative density gives

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

Hence, the probability distribution function of u can be summarized as (2). We notice that the probability of sampling X before Y is the same as the probability of advancing Y before X. This is because the ratio of their occurrence instants is equal to one-half. i.e. $P(u, 1) = 0.5$. Equation (2) is used to calculate the probability of the red team winning the encounter for each interval case as provided in Table 1.