Proceedings of the 1997 Winter Simulation Conference
ed. S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson

SIMULATION OPTIMIZATION USING SIMULTANEOUS REPLICATIONS AND EVENT TIME DILATION

Lee W. Schruben

School of Operations Research and Industrial Engineering
Cornell University
Ithaca, New York 14853, U.S.A.

ABSTRACT

A new approach to simulation response optimization is presented that takes advantage of the ability to run simultaneous replications of different experimental factor settings in a single run. It is also possible to use different time scales for the events corresponding to different design points. In this manner, the run can focus on factor settings that are likely to be optimal and feasible. An example is presented using a penalty function to dilate event times to find the cycle-time constrained capacity of a queue.

1 INTRODUCTION AND BACKGROUND

Simulation experimentation is fundamentally different from conventional real-world sampling. Cost, risk, and the ability to compress time are not the only advantages to experimenting in a simulated environment. It is also possible to favorably influence statistical correlations, concentrate sampling on important events, and continuously observe all aspects of system behavior.

To illustrate some important differences between simulation and real-world experiments, consider a hypothetical situation where we wish to determine which of several automobiles is the fastest. Several drivers, types of tires, and fuel mixtures are at our disposal, each of which might in some way influence the outcome. A particularly crude approach would be to run repeated individual time trials for each car, one at a time, with different combinations of drivers, tires, and fuels. This experiment corresponds to conventional simulation methodology where replications of each combination of factors are run sequentially. Of course, in the real world, it is more efficient to run a series of races, rotating drivers, tires, and fuels according to a reasonable (blocked fractional factorial) experimental design.

2 SIMULTANEOUS REPLICATION

It is possible to do much better than this in a discrete-event simulation experiment. Each event in the simulation program can be assigned a set of parameters that correspond to every factor combination in our experimental design. In effect, we are making implicit copies of the simulation for each design point. All interesting combinations of all drivers, tires, cars, and fuels can then be run simultaneously on a single CPU, picking the best. This experimental strategy is illustrated in Figure 1.

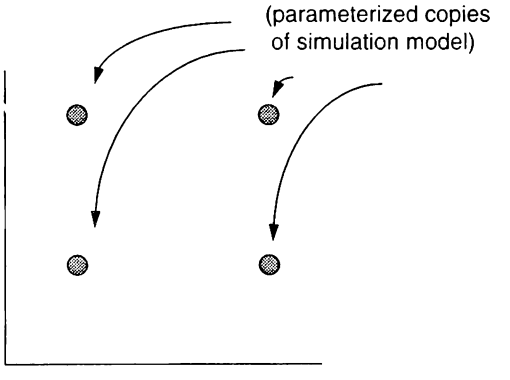

Figure 1: Simultaneous Replication of an Experiment

The concept of simultaneous replication was presented in Section 12.2 of [Schruben, 1995].

Running such a simultaneous experiment might result in an enormous list of future events for the simulation to process. However, only copies of the subset of events that differ for each factor setting need to be parameterized. Furthermore, it is actually possible to take advantage of the larger events list by a technique we will call "time dilation."

3 EVENT TIME DILATION

A characteristic of discrete-event simulations that has not been previously exploited is the ability to scale time during a run. The units measuring time can be changed at any event. We can dilate time differently for the subsets of events corresponding to different experimental points. In this manner, we can cause some events to occur

relatively more often than others. The sets of events with larger time scales will naturally tend to be scheduled near the end of the future events list and occur more rarely. This effectively produces different simulation run lengths for the different points in our experiment (still within a single execution of the model). By controlling time dilation, we can concentrate the simulation run on interesting factor settings in our experimental design. Unlike sequential replications, initialization bias is taken care of once for all points in the experiment. The concept of time dilation was presented as homework 4.3.3 in [Schruben, 1995].

Time dilation during a discrete-event simulation run does not cause serious difficulty in collecting valid output statistics. Since the system state and the time scales can only change at events, it is a trivial matter to adjust time-integral statistics. All output time series from a discrete-event simulation are step functions; at each event, the area under each step is simply divided by the current time scale.

As an aside: Sharing common exogenous events in a simultaneously replicated experiment is an ideal way to implement the variance reduction technique of using common random streams without having to worry about synchronization. For example, a common input event can be used to drive factory simulations with different scheduling rules or tool configurations.

4 ESTIMATING CYCLE-TIME CONSTRAINED QUEUE CAPACITY

As a tangible example, consider an important problem in semiconductor manufacturing. Estimating the 4X capacity of a semiconductor fab. Specifically, for a given tool set, this is the maximum rate at which wafers can be released to the floor such that the queueing delay for an average wafer does not exceed 4 times the theoretical minimum cycle-time. (If a mix of products is being run, an average wafer flow time is used.) It is easy to estimate this theoretical minimum cycle-time using simulation; just run one wafer through the system (without tool failures), measure its time, and replicate. The 4X fab capacity is defined in Figure 2.

For an actual fab the cycle-time/release-rate trade-off curve can be considerably more complicated. This curve will initially decrease for batch tools when the release rate is low and the tool must wait for its minimum batch size (it may be possible for every batch tool in a fab to produce its own local minimum in the curve). Random factors such as unplanned tool down-time will make the curve steeper. The situation is complicated greatly by the fact that the asymptotic release rate, above which WIP growth becomes unstable, is not known. Engineering lots, test wafers, hot lots, and scheduled tool maintenance can tend to decrease this upper bound while yield loss tends to move it to the right.

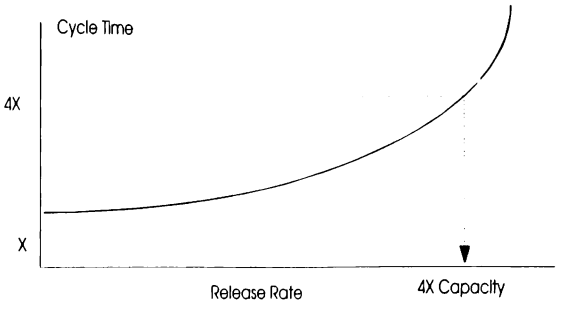

Figure 2: Estimating the 4X fab Capacity

Determining the cycle-time constrained fab capacity is straightforward when the cycle-time function is known from queueing theory. However, small errors in approximating this trade-off curve can cause large errors in capacity estimation with potentially severe economic consequences to the enterprise. This is particularly true for the short competitive cycle-times found in the semiconductor industry; the trade-off curve in this region is flat.

There are even more serious problems if simulation must be used to estimate fab capacity. Conventional simulation experiments involve making a series of runs trying different release rates until one giving approximately the target average cycle time is found. Sometimes stochastic approximation or regression techniques are helpful [Chance and Schruben, 1995].

Standard simulation experiments are almost useless in estimating the asymptotic upper bound on fab capacity. At high release rates, observed cycle times are highly correlated and run initialization bias is a serious concern. These factors combine to give simulation estimators of heavy-traffic cycle times both high bias and high variance. Queueing theory was used with some success to estimate this capacity bound in the SEMATECH sponsored MIMAC study. However, simulation of millions of wafer flows were needed to estimate 4X capacities.

A novel experimental strategy for simulation optimization under current development is presented next. Here the release event driving the simulation is assigned parameters from a grid of release rates. A range of release rates can then be simultaneously simulated during a single run. This grid can be refined during the run for promising experimental regions as is done in the variance reduction technique of splitting.

As the simulation progresses we want to spend more and more time running events that are near the solution. To do this we will penalize rates that are not

performing well or appear to be infeasible. The "penalty function" takes the form of time dilation for events associated with release rates that are unlikely to be near the 4X capacity. When their relative time scales are increased, events will naturally tend to be scheduled near the end of the future events list. If the events list is very large, these penalized events become essentially irrelevant and have no detrimental effect on execution speed. The number of event executions devoted to a particular design point reflects the likelihood that the design point is optimal. Hence, the simulation run is concentrated on those experimental points where success is most likely. This has the positive effect of minimizing the estimator variance at exactly the right place.

Events corresponding to uninteresting parameter values will occur occasionally as in simulated annealing. This is necessary if there is to be any theoretical hope of global optimization.

For illustration: this experimental technique was tested with a simulated M/M/1 queue. The simultaneous replication strategy and time dilation techniques in [Schruben, 1995] were used. For this test system, the true 4X capacity rate is known. To put this problem in perspective, conventional replication methods would require hundreds of thousands of simulated jobs to get a reasonably good estimate of the cycle-time constrained capacity of this simple system.

We will set the average processing time at 1. From queueing theory it is known that the 4X capacity for this system is .8 and the maximum stable release rate is 1. Without using any of this information (or even the fact that cycle times increase with release rate), we will estimate the 4X capacity of this system with one simulation run. To do this within an error of 1/24, a grid of 40 release rates from .03125 to 1.25 is run. Recall that (presumably unknown to us) rates over 1 will make the system unstable.

Let W_i denote the average job delay corresponding to the ith release rate. For time dilation, the time scale for the release event at rate i was multiplied by $(W_i\text{-}4)^P$. In this grid of 40 rates, the 4X capacity corresponds to event parameter i=24, which has a release rate of .8. All event indices over 30 correspond to unstable release rates.

After an initialization period of 5,000 jobs, the relative frequency of events corresponding to the different release rates appear in Figure 3a.

For the next 5,000 jobs, a quadratic time dilation penalty was then invoked. After a total of 10,000 simulated jobs shared across all 40 systems, the event index at the correct solution of 24 (capacity of .8) is clearly indicated in Figure 3b. Events with unstable release rates received early penalties, effectively excluding them from the experiment.

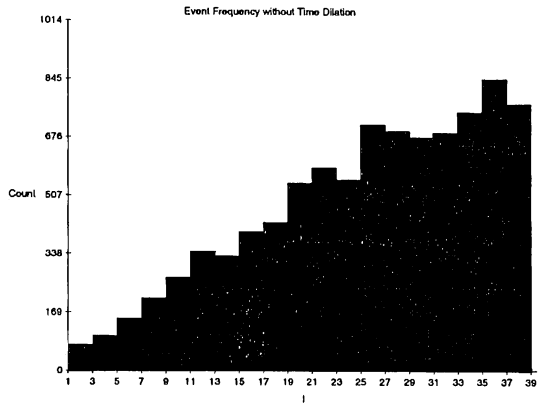

Figure 3a: Event Execution Frequency without Time Dilation Penalties

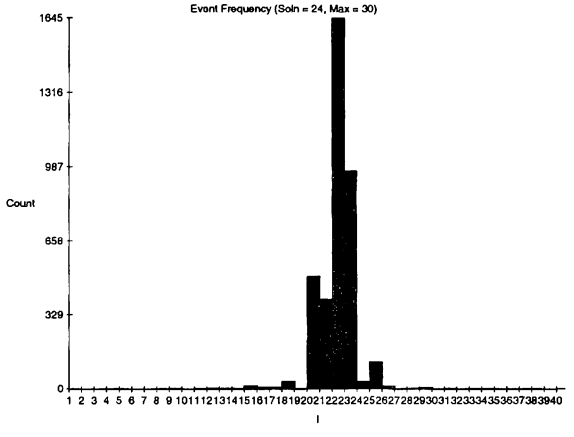

Figure 3b: Event Execution Frequency with Time Dilation Penalties

In Figures 4a and 4b, average event indices are plotted for both the 2X and 4X capacities. Periodically, these averages are reset. These averages quickly converged to the correct indices of 20 (capacity=.66) and 24 (capacity=.8).

The development of advanced simulation experimental techniques requires advancements in queueing theory. It is vital that analytical cycle-time/release-rate tradeoff curves be developed for queueing models that more closely represent actual semiconductor manufacturing tool and fab behavior. It is not possible to assess the validity of simulation problem solution methods without queueing models providing a test bed for which the correct answers are known.

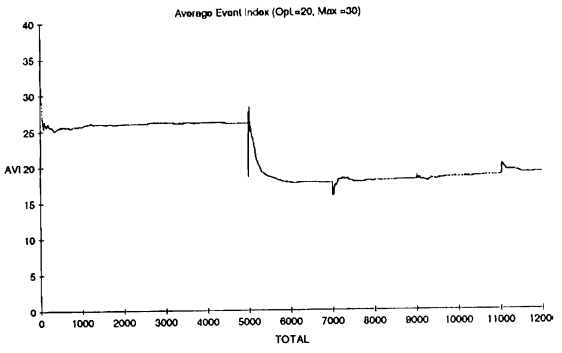

Figure 4a: Average Event Indices for a 2X Capacity Experiment

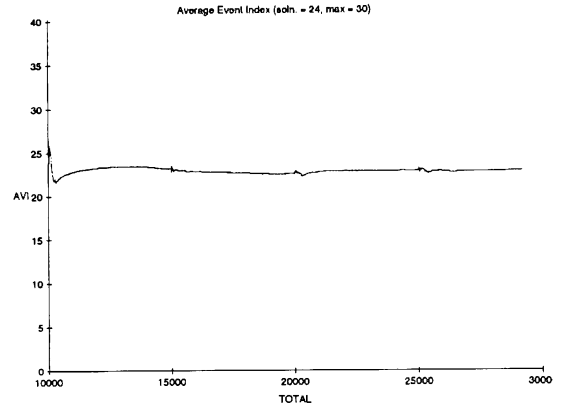

Figure 4b: Average Event Indices for a 4X Capacity Experiment

5 COMMENTS AND TOPICS FOR FURTHER RESEARCH

The example of time dilation presented here is only one of many ways to take advantage of the abilities to run simultaneous replications and scale time in discrete-event simulation experiments. It may be better to have the events for unpromising experimental factor settings skip intervals of time altogether. The events for each factor setting would then tend to fall in time buckets on the future events list. Global optimization methods could be developed by revisiting unpromising factor settings less and less frequently as the run progresses. The output time series could be stitched together again illustrating the advantages of a "stitch in time."

ACKNOWLEDGMENT

The research reported here was partially supported by NSF Grant DMI-9322712 and a joint SRC and NSF research project in semiconductor operations modeling.

All examples were graphically coded in C by SIGMA for Windows [Schruben, 1995].

REFERENCES

Chance, F., and L. Schruben. 1995. Estimating cycle-time constrained capacity via simulation. *IIE Research Conference Proceedings*, Los Angeles.

Schruben, L. 1995. *Graphical Simulation and Modeling using SIGMA for Windows*, 3rd Edition. The Scientific Press.

AUTHOR BIOGRAPHY

LEE SCHRUBEN is a Professor at Cornell University's School of Operations Research and Industrial Engineering. He received his undergraduate degree in engineering from Cornell and a Ph.D. from Yale. He also has a Master's Degree from the University of North Carolina. His research interests are in statistical design and analysis of simulation experiments and in graphical simulation modeling methods. His simulation application experiences and interests include semiconductor manufacturing, dairy and food science, health care, military, and the banking and hospitality industries.