Proceedings of the 2005 Winter Simulation Conference
M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.

SHOULD TRANSIENT ANALYSIS BE TAUGHT?

M. C. Court
J. L. Pittman
H. T. L. Pham

School of Industrial Engineering
202 West Boyd, Room 124
University of Oklahoma
Norman, OK 73019, U.S.A.

ABSTRACT

The authors present the results of experiments performed to identify the pitfalls of performing 'bad' transient analysis when estimating steady-state parameters via the method of independent replications. The intention was to demonstrate to students that failure to delete transient data may lead to confidence intervals that underestimate steady-state parameters. Two types of systems are analyzed: M/M/1/GD/∞/∞ systems and an M/M/s/GD/∞/∞ optimization problem. These systems are chosen since they are typically taught in an undergraduate stochastic operations research course where a closed-form solution of the steady-state parameter exists. Surprisingly, the results prove to support the opposite of our original intention—regardless of run length, ignoring transient analysis often leads to the same level of coverage at greater precision, or provides no gain in coverage to justify the effort of performing transient analysis. Thus, we now pose the question—should transient analysis be taught?

1 INTRODUCTION

One of the most difficult topics for undergraduate students taking their first stochastic operations research or simulation course is non-terminating systems analysis (Court 2001). At issue is the student's ability to understand that output data generated from a non-terminating system's simulation will have both transient and steady-state data (i.e., the data are not iid—independently and identically distributed data). Additionally, they are uncomfortable or inexperienced with utilizing approximation tools (simulation) that rely on ad-hoc methodologies (e.g., graphical techniques to distinguish between transient and steady-state behavior) and statistical laws (e.g., the central limit theorem) for parameter estimation. This is understandable, since more often than not, undergraduate students have only used mathematical modeling techniques that are 'guaranteed' (as long as the underlying assumptions of the technique are not violated) to generate 'one-and-only' (or hopefully, the optimal) solution to a problem. In contrast, simulation output analysis for non-terminating systems may generate several different 'approximations' for the unknown parameter of interest.

Complicating the issue is the inability of the students to validate their results. In general, these students are not comfortable with the amount of judgment/skill/experience required to evaluate their findings (e.g., confidence intervals about the parameters of interest). Simulation analysis (particularly simulation output analysis of non-terminating systems) tends to be 'too ad-hoc' for the 'typical' undergraduate student; while non-terminating output analysis is too important a topic to ignore when teaching simulation modeling courses. In our opinion, ignoring this topic when teaching simulation modeling is equivalent to generating 'inadequate simulation practitioners'.

We present a set of experiments to help identify the pitfalls of performing 'bad' transient analysis when estimating steady-state parameters via the method of independent replications. The intention of the experiments is to demonstrate to industrial engineering undergraduate students that failure to delete transient data (or not enough transient data) will lead to 'poorer' confidence intervals than confidence intervals generated when the student takes the time to more accurately identify and discard transient data. In fact, our goal is to demonstrate that these confidence intervals are more likely to not cover the true mean, or precision (half-width size) will be less (greater half-width size) than the confidence intervals generated when transient data are discarded. Our experiments are designed for two cases of non-terminating systems where we will perform what we define as 'perfect transient analysis'. While performing the experiments, we will also explore the impact on confidence interval generation when the analyst has run the simulation long enough to generate what we define as 'perfect run length' versus not running the simulation long enough—'insufficient run length'. The results are surprising: for the cases where transient analysis is done 'badly' (or even ignored), the confidence intervals provide coverage at greater precision than those cases where 'perfect transient analysis' is performed.

The remaining sections of the paper are in the following order:

- The definitions section delineates what we mean by performing transient analysis 'badly' and presents our definitions of perfect transient point, worst-case transient analysis and perfect run length.
- The methodology, results, and analysis section defines the experiments we performed, how our definitions were put into practice and the resulting confidence intervals.
- The conclusions and future research section is a discussion of the results, some questions we pose for educators and the avenues of future research we wish to explore.

2 DEFINITIONS

The goal of the experiments is to demonstrate the impact on confidence intervals (generated via the method of independent replications) when the analyst performs transient analysis 'badly'. First, we need to agree on how 'bad transient analysis' should be defined; and to offset this definition, we need to agree on how 'perfect transient analysis' should be defined. Additionally, we need to develop definitions that can be understood at the undergraduate level.

We propose that a student is able to achieve *'perfect transient analysis'* when s/he identifies the point in the output data such that from that point to the end of the simulation run, the average of the remaining data equals the true mean of the unknown parameter of interest. For example, if the student wishes to obtain an estimate of the true average waiting time in queue, 'perfect transient analysis' equates to deleting enough of the initial data, such that the remaining data, when averaged (the sample mean from the data), equals the true mean.

To investigate cases of performing transient analysis 'badly', we agreed that the *'worst-case-transient analysis'* is to have the student/practitioner ignore (either intentionally or unintentionally) transient analysis altogether. In other words, we need to investigate cases where the student/practitioner does not delete any initial data from the simulation runs and hence, the method of independent replications would see its worst case of initialization bias.

However, we also wish to explore cases where the student/practitioner 'just happens to' overcome the initialization bias of the worst-case-transient analysis by generating 'enough' steady-state data to offset the initialization bias. So, we will explore cases where even though the practitioner intentionally or unintentionally chose to ignore transient analysis, s/he still generated what we call the 'perfect run length'. Then, *'perfect run length'* is defined as the ability of the analyst to intentionally or unintentionally ignore transient analysis and yet, generate a simulation run whose data, when averaged across the entire run (the replication's sample mean), equals that of the true mean.

Note that while several transient analysis methods exist (see Law and Kelton 2000) we will use a pilot run for determining transient analysis and will apply that warm up period (perfect transient point) across all runs. Then, for the method of independent replications, transient deletion in the remaining runs (replications) will be treated as a constant (perfect transient point of the pilot run).

Utilizing a pilot run is also the approach we will take for analyzing perfect run length. In practice, the run length is highly influenced by the objectives of the simulation study and is typically determined via a pilot run of the simulation (and usually after transient analysis has been concluded). So again, as with the ad-hoc methodologies of transient analysis, run length selection is dependent on the analyst's ability to 'judge' how long the simulation should be run. However, once run length is determined, it is fixed to uphold the method of independent replications requirement of fixed run lengths for all runs. Thus, for our perfect run length cases, we will generate a pilot run of the model to determine the time of perfect run length and then, run the remaining simulations with the perfect run length time invoked as the stopping rule for all runs.

3 METHODOLOGY, RESULTS, AND ANALYSIS

Two queuing systems cases are analyzed:

- **Case 1:** $M/M/1/GD/\infty/\infty$ systems at three levels of ρ (=0.50, 0.75, 0.90).
- **Case 2:** An $M/M/s/GD/\infty/\infty$ optimization problem with λ=2/minute, μ=0.5/minute, a per server cost of $9/hour and a delay cost to the customer of $0.05/minute, at s=5 and s=6 (see Winston 2004).

The cases are chosen since well-known queuing theory results exist, and these systems are typically introduced in an undergraduate stochastic operations research course. The parameter of interest for Case 1 is the average waiting time in queue. The Case 2 system types are characteristically used to introduce the formulation required to solve queuing optimization problems. Here, the objective is to minimize the total expected cost in terms of the expected delay cost to the customer as a function of the service level (the number of servers, s). In queuing analysis, the formulation is straight-forward (see Winston 2004) and a closed-form solution exists for the minimum expected total cost. For the given values of our Case 2 system, the minimum total expected cost occurs when s=5. Since the waiting cost to the customer and the per server costs are fixed, to solve the problem through simulation analysis is equivalent to determining a parameter estimate for the average waiting time in queue (W_q—the only unknown parameter). All cases were simulated via the Arena 7.01 software (Kelton et al. 2004) and analyzed via Arena 7.01 and Microsoft® Excel©.

For the systems of Case 1, the methodology is as follows:

1. A pilot run of the model is made for the waiting time in queue where the stopping rule for the pilot run is invoked when the average waiting time across the run equals the theoretical value. At this point the simulation is terminated and the simulation's run length is noted as the **perfect run length**. Then, 20 independent replications of the simulation model are generated with the perfect transient's run length time utilized as the stopping condition for each of the 20 runs. The mean from each of the replications is then used to generate a 95% confidence interval about the average waiting time in queue. See Table 1 for a list of the perfect run lengths identified in the experiment.
2. Run lengths of 6,000; 20,000; 50,000; and 100,000 time units are utilized for all cases to identify the perfect transient point at various run lengths and to allow the **worst-case transient analysis** (no deletion of transient) at various run lengths to be explored. Since the perfect run length determined for ρ=0.90 was found to be greater than 100,000 time units, an additional case is utilized at a run length of 1,000,000 time units (see Table 2).
3. For all run lengths of 2 above, (except perfect run length), perfect transient analysis is invoked to determine where in that particular run length the average of the data equals the theoretical value (**perfect transient point**). The process is to export the output data of the simulation's pilot run into a Microsoft® Excel© spreadsheet and perform what one might call a *'reverse cumulative average'*. For example, with a run length of 20,000 time units, the first average obtained is of all of the output responses generated over the 20,000 time units. If this average equals the true mean, the entire run is considered in steady-state. If not, the first observation is dropped and the remaining data are averaged. If the remaining data's sample mean equals that of the true mean, then the simulation time for the second observation is noted as the **perfect transient point**. If not, the second observation is dropped from the data set and the process of dropping each successive observation is repeated until the perfect transient point is found. See Table 2 for a list of the perfect transient points. Note, that for some run lengths, perfect transient can not be found. For the run lengths of 20,000 time units and 50,000 time units, the perfect transient points for ρ=0.50 are considerably greater than the perfect transient points for ρ=0.75. Also note that for the run length of 100,000 time units, the perfect transient point is 0.00 for ρ=0.50 and 81,009.49 for ρ=0.75. At first this may seem to contradict the trend expected for longer run lengths within a particular ρ or for the same run length across different ρ's. However, an explanation can be found by the fact that only a pilot run of the simulation is used to determine the perfect transient point; transient itself is stochastic and thus, the perfect transient point is also stochastic. So while we would expect to see the perfect transient point increase as run lengths increase, since it is stochastic, it will 'move' within a ρ at various run lengths and 'move' for run lengths at various ρ's.
4. If a perfect transient point can be found for a particular run length, two more confidence intervals are generated via the method of independent replications:
   (a) First, when the replications have the perfect transient deleted but the total run length is terminated at the original run length's time unit. For example, if the run length is 20,000 time units and the perfect transient point is

Table 1: Worst-Case Transient Analysis Run Lengths and Perfect Run Lengths for Case 1 Systems

Run Length	ρ=0.50	ρ=0.75	ρ=0.90
Perfect Run Length	62,314.95	16,709.11	965,052.03
Worst-Case Transient Analysis	6,000	6,000	6,000
	20,000	20,000	20,000
	50,000	50,000	50,000
	100,000	100,000	100,000
			1,000,000

Table 2: Perfect Transient Point of Run Lengths for Case 1 Systems (-- indicates not found)

Run Length	ρ=0.50	ρ=0.75	ρ=0.90
6,000	--	--	--
20,000	12,210.40	868.93	--
50,000	28,675.29	672.31	--
100,000	0.00	81,009.49	--
1,000,000			250,537.13

found to occur at 12,210 time units, each replication will have a total of 7,790 time units worth of data available to calculate each replication's sample mean.

(b) Secondly, when perfect transient is deleted from each of the replications and the total run length is modified to equal that of the perfect transient's time units plus the original run length's time units. Following the same example in (a), the new run length is 32,210 time units for each of the replications, where the first 12,210 time units are specified as 'warm-up' (amount of simulation time deleted as transient) and the remaining 20,000 time units of data are available for calculating each replication's sample mean.

For Case 2, the methodology for Case 1 is invoked for two realizations of s=5 and s=6. Table 3 contains the prefect run length and worst-case transient analysis run lengths for the Case 2 systems. Note, an additional run length of 500,000 time units is required when s=6 since perfect run length is found at 310,498.33 time units. Table 4 contains the perfect transient point of the run lengths, and as with some of the Case 1 systems, some run lengths for the Case 2 systems contain no perfect transient point.

Table 3: Worst-Case Transient Analysis Run Lengths and Perfect Run Lengths for Case 2 Systems

Run Length	s=5	s=6
Perfect Run Length	11,218.90	310,498.33
Worst-Case Transient Analysis	6,000	6,000
	20,000	20,000
	50,000	50,000
	100,000	100,000
		500,000

Table 4: Perfect Transient Point of Run Lengths for Case 2 Systems (-- indicates not found)

Run Length	s=5	s=6
6,000	--	1251.18
20,000	--	14,400.08
50,000	47,917.48	48,129.71
100,000	76,131.37	33,915.82
500,000		315,533.26

Table 5 reveals the 95% confidence intervals generated for the worst-case transient analysis and perfect run lengths (as defined in Table 1) of the Case 1 systems. A shaded box indicates a run length that does not apply for that ρ. For each ρ, the confidence interval generated at perfect run length is indicated by a double-border cell. Of the

Table 5: 95% Confidence Intervals of Worst-Case Transient Analysis Run Lengths and Perfect Run Lengths for Case 1 Systems

M/M/1	ρ=0.50	ρ=0.75	ρ=0.90
W_q / Run Length	0.500	2.250	8.100
6,000	0.507+/- 0.019	2.363+/- 0.095	8.136+/- 0.810
16,709.11		2.280+/- 0.056	
20,000	0.499+/- 0.008	2.262+/- 0.050	8.229+/- 0.446
50,000	0.500+/- 0.007	2.260+/- 0.042	7.962+/- 0.230
62,314.95	0.500+/- 0.004		
100,000	0.501+/- 0.003	2.277+/- 0.028	8.109+/- 0.176
965,052.03			8.077+/- 0.070
1,000,000			8.083+/- 0.069

16 confidence intervals generated, only one failed to contain the true mean (W_q). This occurs at the 6,000 run length for ρ=0.75. Note, this is the only confidence interval generated by a run length below perfect run length that does not contain W_q. Recall that for all of the run lengths, no transient is deleted; yet, 15 of the 16 confidence intervals contain the true mean. For ρ=0.50 and ρ=0.75, the best precision for the confidence interval is attained at the 100,000 run length; while for ρ=0.90 it is attained at the 1,000,000 run length. These results indicate that there is no advantage to attaining perfect run length in terms of generating a confidence interval that contains W_q. However, the analysis does show that as the run length increases, the precision of the confidence interval improves.

Table 6 contains the 95% confidence intervals generated from the perfect transient analysis runs of the Case 1 systems. Recall from the methodology section that two types of run lengths are performed after the perfect transient point is identified. The first, (a), is run at the original run length with a warm-up period set at the perfect transient point. Thus, the total simulated time is the original run length, but the output data collected from each replication have the transient data deleted from the run. The second, (b), is run with the total simulated time equal to the original run length plus the perfect transient point. The net effect is that (b) will have 'steady-state' data collected for the original run length's time units while (a) will have less 'steady-state' data collected. A '--' symbol indicates that a perfect transient point could not be found for the run length of (a) (see Table 2). A shaded box indicates a run length that does not apply for that ρ. All confidence intervals

Table 6: 95% Confidence Intervals Generated via Perfect Transient Analysis at Two Run Lengths for Case 1 Systems: (a) Original Run Length and (b) Original Run Length + Perfect Transient Point

M/M/1 / W_q / Original Run Length	ρ=0.50	ρ=0.75	ρ=0.90
	0.500	2.250	8.100
6,000 (a)	--	--	--
(b)	--	--	--
20,000 (a)	0.488+/-0.012	2.264+/-0.054	--
(b)	0.496+/-0.013	2.266+/-0.057	--
50,000 (a)	0.500+/-0.005	2.261+/-0.044	--
(b)	0.511+/-0.005	2.257+/-0.044	--
100,000 (a)	0.501+/-0.003	2.289+/-0.050	--
(b)	0.501+/-0.003	2.269+/-0.023	--
1,000,000 (a)			8.087+/-0.094
(b)			8.112+/-0.076

generated contain W_q; and, except for two sets of confidence intervals (i.e., ρ=0.75 with original run length at 100,000 and ρ=0.90 with original run length at 1,000,000), the precision of the (a) confidence intervals is the same or better than the precision of the confidence intervals of (b). Then, there seems to be no advantage to increasing the run length for 'steady-state' data collection once perfect transient is deleted. And, as with the previous results of Table 5, precision improves as run length increases.

Tables 7 and 8 reveal the results of the analysis for the Case 2 systems. As with the Case 1 systems, the same trends exist for the Case 2 systems' confidence intervals:

- For the worst-case scenario and perfect run lengths (see Table 7), there is no advantage to attaining perfect run length in terms of generating a confidence interval that contains W_q; and as the run length increases, the precision of the confidence interval improves.
- For the perfect transient runs (see Table 8), there seems to be no obvious advantage to increasing the run length for 'steady-state' data collection once perfect transient is deleted; and, as with all previous results, precision improves as run length increases.

Table 7: 95% Confidence Intervals of Worst-Case Transient Analysis Run Lengths and Perfect Run Lengths for Case 2 Systems

M/M/s / W_q / Run Length	s=5	s=6
	1.108	0.285
6,000	1.140+/-0.092	0.294+/-0.026
11,218.90	1.154+/-0.054	
20,000	1.138+/-0.041	0.298+/-0.013
50,000	1.117+/-0.026	0.290+/-0.007
100,000	1.114+/-0.018	0.286+/-0.005
310,498.33		0.286+/-0.002
500,000		0.286+/-0.002

Table 8: 95% Confidence Intervals Generated via Perfect Transient Analysis at Two Run Lengths for Case 2 Systems: (a) Original Run Length and (b) Original Run Length + Perfect Transient Point

M/M/1 / W_q / Original Run Length	s=5	s=6
	1.108	0.285
6,000 (a)	--	0.299+/-0.028
(b)	--	0.299+/-0.026
20,000 (a)	--	0.288+/-0.012
(b)	--	0.288+/-0.015
50,000 (a)	1.102+/-0.095	0.283+/-0.007
(b)	1.101+/-0.103	0.282+/-0.008
100,000 (a)	1.104+/-0.016	0.282+/-0.004
(b)	1.103+/-0.018	0.285+/-0.004
500,000 (a)		0.284+/-0.002
(b)		0.284+/-0.002

4 CONCLUSIONS AND FUTURE RESEARCH

Granted, the confidence intervals generated are for only one set of 20 replications each, while coverage analysis requires several sets of 20 replications to predict the accuracy of confidence intervals' coverage (see Law and Kelton 2000); but this is not the intention of the experiments. The experiments are to convey to undergraduate students the danger of not performing transient analysis when estimating an unknown parameter for a non-terminating system. We chose the method of independent replications since it tends to be more readily understood by undergraduate students, then say, the batch means method (see Court 2004). Additionally, the method (independent replications) tends to suffer from initialization bias and thus, performing transient analysis should help to reduce the bias.

Since transient analysis relies on ad-hoc techniques, students tend to have difficulty in applying the techniques and justifying/validating the results (see Court 2004). We chose two types of queuing systems that are typically taught in a stochastic operations research course to illustrate transient analysis issues. Surprisingly, we found very little evidence to support the fact that performing transient analysis will lead to 'better' confidence intervals. That is, even when transient analysis is ignored, the true mean is contained within the confidence interval and usually with greater precision. Table 9 also supports this finding. Here, we compare 'equivalent' run lengths for the Case 1 systems by restating the results found in Tables 5 and 6. For the (b) confidence intervals of Table 6, the total run length is the original run length plus the time unit of the perfect transient point. So, the (b) confidence intervals have the same amount of data collected (in terms of time) over the run as the worst-case transient analysis run length times of Table 5. However, recall that the worst-case transient analysis occurs when no transient is deleted. So, in theory, the (b) generated confidence intervals should be 'better' than the confidence intervals generated under the 'equivalent' worst-case transient analysis run length, since they contain 'steady-state' data. Also recall, and as displayed in Table 9, that a '--' symbol indicates that a perfect transient point could not be found for the run length and only the (*) confidence interval did not contain W_q. Then, in general, the 'best' half-width occurs when no transient is deleted. In fact, the worst-case transient analysis runs generated more 'valid' (W_q is within the confidence interval) confidence intervals at shorter run lengths than (b) (since perfect transient could not be found at the shorter run lengths). So, the runs with no transient deleted generated 12 valid confidence intervals, while the 'steady-state' runs only generated 7 valid confidence intervals. For the 7 valid confidence intervals generated via the 'steady-state' runs, the runs with no transient deleted generated confidence intervals with equal or better precision 5 out of those 7 times. The same comparison (see Table 10) when performed for the Case 2 systems yields similar, if not better, results.

Table 9: 95% Half-Widths of Worst-Case Transient Analysis versus Perfect Transient Analysis at 'Equivalent' Run Lengths for Case 1 Systems

'Equivalent Run Length'	Worst-Case Transient Analysis (see Table 5)	(b) Run Length (see Table 6)
ρ=0.50		
6,000	0.019	--
20,000	0.008	0.013
50,000	0.007	0.005
100,000	0.003	0.003
ρ=0.75		
6,000	0.095*	--
20,000	0.050	0.057
50,000	0.042	0.044
100,000	0.028	0.023
ρ=0.90		
6,000	0.810	--
20,000	0.446	--
50,000	0.230	--
100,000	0.176	--
1,000,000	0.069	0.076

Table 10: 95% Half-Widths of Worst-Case Transient Analysis versus Perfect Transient Analysis at 'Equivalent' Run Lengths for Case 2 Systems

'Equivalent Run Length'	Worst-Case Transient Analysis (see Table 7)	(b) Run Length (see Table 8)
s=5		
6,000	0.092	--
20,000	0.041	--
50,000	0.026	0.103
100,000	0.018	0.018
s=6		
6,000	0.026	0.026
20,000	0.013	0.015
50,000	0.007	0.008
100,000	0.002	0.004
500,000	0.002	0.002

Thus we pose the question: At the undergraduate level, should transient analysis be taught when teaching the method of independent replications? Our conclusion is 'yes' but with the following emphasis:

- Transient analysis is an ad-hoc methodology and as such, there is no guarantee that the confidence interval generated will be 'better' (greater precision) than a confidence interval generated when transient data are present.
- Run length seems to be the most important factor when trying to obtain confidence intervals containing the true mean.

Our opinion is that strong emphasis should be placed on the run length. It seems that, at least, for the systems studied, transient data whether present or not, have very little impact on the validity of the confidence interval; run length however, does. So the student should come away with the realization that running the simulation 'long enough' is much more important than identifying steady-state behavior.

Our confidence intervals were generated via the method of independent replications. Future research will be aimed at the impact of transient analysis when utilizing the batch means method. Our prediction is that since the batch means method is a single-replication method (see Law and Kelton 2000 or Kelton et al. 2004), determining the perfect transient point will assist in providing a better confidence interval than if transient analysis is performed 'badly'. We are not comfortable with the same prediction for perfect run length analysis since if independent batches can be generated, at least one of the batches will contain transient data in the sample (or batch) mean.

REFERENCES

Court, M. C. 2001. A case study on the impact of web-based technology in a simulation analysis course. *Simulation, Special Issue: Simulation in Education and Education in Simulation:* 76(4). 207-213.

Court, M. C. 2004. The impact of using Excel macros for teaching simulation input and output analysis. *International Journal of Engineering Education:* 20(6). 966-973.

Kelton, W. D., R. Sadowski, and D. Sturrock. 2004. *Simulation with Arena, Third Edition*. McGraw Hill, Boston, MA.

Law, A. and W. D. Kelton. 2000. *Simulation Modeling and Analysis, Third Edition.* McGraw Hill, Boston, MA.

Winston, W. 2004. *Operations Research: Applications and Algorithms*. Brooks/Cole-Thomas Learning, Belmont, CA.

AUTHOR BIOGRAPHIES

MARY COURT is an Associate Professor in the School of Industrial Engineering at the University of Oklahoma. She has taught statistical analysis of simulation and simulation modeling courses for over 10 years. Her research interests lie in simulation output analysis and electronic education. She has received several grants for research in education and publishes in that area. Her research has been funded by the Federal Aviation Administration, the United States Air Force, the National Science Foundation and the Department of Transportation. She is a member of IIE. Her email address is mcourt@ou.edu.

JENNIFER PITTMAN is a doctoral student at the University of Oklahoma. She received her B.S and M.S. in the accelerated IT program of the School of Industrial Engineering with minors in computer science and philosophy. Her masters thesis was focused on modeling techniques for developing facility layout designs. She is an NSF Graduate Research Fellow and her doctoral research focuses on developing computer simulated environments for analyzing human movement under life threatening conditions. Her e-mail address is jlpittman@ou.edu.

HUONG PHAM is currently working on her masters at the University of Oklahoma. She is a teaching assistant at the School for the undergraduate stochastic operations research and simulation modeling courses, as well as the graduate level statistical analysis of simulation and modeling courses. She received her B.S. in the accelerated IT program of the School of Industrial Engineering. She is the recipient of the Cleo Cross International Student Scholarship. Huong's masters thesis is in the area of simulation optimization. Huong plans on defending December 2005 and entering the doctoral program at the School in January 2006. Her email address is lanhg@ou.edu.