

A SIMULATION COURSE FOR HIGH SCHOOL STUDENTS

David Goldman

H. Milton Stewart School of
Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332, U.S.A.

ABSTRACT

Computer simulation presents a variety of opportunities for high school students to receive exposure to mathematics and engineering in the real world. We describe in a high-level way a course that uses computer simulation to enhance students' general modeling skills in probability and statistics, queueing models, financial engineering, and programming. Our experience has been that students can easily handle the material, and certainly seem to enjoy the experience.

1 INTRODUCTION

One of the most important goals in secondary education today is that of enhancing the mathematics abilities of high school students. An obvious way to go about improving students' appreciation of mathematics is to incorporate exciting real-world applications into mathematics curricula; and applications taken from operations research, industrial engineering, and management science seem to be ideal for the job.

Computer simulation presents a wonderful variety of opportunities for high school students to receive exposure to mathematics and engineering in the real world. Indeed, we encounter potential simulation applications in common activities from everyday life; for example,

- Queuing at the bank,
- Waiting in the grocery store line,
- Planning a day's activities (and then waiting in line at) Disneyworld,
- Dealing with plane travel routing, paying for plane fare, and then waiting for the pilot and crew to show up,
- Financial planning — how to spend or save earnings,
- Resource allocation problems, for instance, at a hospital.

In this short paper, we describe in a high-level way a course that uses computer simulation to enhance students' general modeling skills in probability and statistics, queueing models, financial engineering, and programming. Our experience has been that students can easily handle the material, and certainly seem to enjoy the experience.

The rest of this article proceeds as follows. In Section 2, we propose a strawman syllabus, Section 3 gives some details on the components of the syllabus, along with a few simple examples, Section 4 describes our relevant experiences with high school students and teachers, and Section 5 gives some conclusions and ideas for future work.

2 A SAMPLE SYLLABUS

The overall theme of any simulation is to characterize the behavior of a system (whether it is real or merely in the planning stages) that may have a stochastic component. For instance, consider a simple static Monte Carlo experiment, such as estimating the probability of a certain event — e.g., what is the probability that there will be two kids in the classroom having the same birthday? For such problems, we might be interested in thinking about questions like

- What is “the answer” — i.e., what is the probability that there will be a common birthday?
- If we run a simulation, how many kids will it take before we get a match?
- What is the variability inherent in this experiment?

In dynamic queueing models, we might be interested in studying more-sophisticated questions, including

- What will be the throughput?
- How can we change it?
- Where are the bottlenecks?
- What is the best design?
- What is the reliability of the system?
- What is the impact of breakdowns?

With these motivational goals in mind, one can structure a strawman syllabus, which can serve as the basis for a semester-style course or even a short course for high school students. For example, we have used the following outline in the past for full and short courses given to high school teachers and students.

- General background
- Introduction to probability and statistics
- Hand simulations
- Spreadsheet simulation
- Random variate generation
- Input modeling
- Output analysis
- A discrete-event simulation language

All components of the syllabus can be based on a series of everyday examples, which can be carried out by hand, by Excel, or via a standard simulation language such as Arena, ProModel, Extend, etc.

3 SOME DETAILS AND EXAMPLES

Now we give some details with respect to some of the topics outlined above.

3.1 General Background

One can begin the background discussion by briefly describing what operations research is, and how simulation fits into the discipline. We usually start off the simulation content with some simple motivational examples involving static Monte Carlo, and then progress to more-complicated, more-impressive-looking simulation demonstrations from real-world applications. These demonstration models serve to convey to the students that “this is what you can potentially do with a little practice” — after all, the idea here is to keep things interesting for the students from the onset.

Some of the basic motivational examples include simulation of the classic birthday problem, an easy method to estimate π , some illustrative random variate generation techniques (with pretty pictures), etc. See the introductory tutorial Goldman (2007) in this *Proceedings* for some specifics. The real-world applications include simulations of an immunization clinic, an airline’s flight routing structure, and the propagation of a disease in Africa.

3.2 Introduction to Probability and Statistics

Even though a high school simulation course does not need to be calculus-based, today’s good high school students may have already had (or will be concurrently taking) Advanced Placement courses in calculus and/or statistics. The point is that many high school students will be able to handle

a gentle primer on probability and statistics — certainly enough to be able to handle the corresponding needs of our simulation course.

Useful topics include: elementary counting techniques, conditional probability, simple discrete and continuous random variables, with a little additional emphasis on the normal distribution, descriptive data analysis, and at least some time spent on confidence intervals for unknown parameters. Even though the list of topics may appear at first glance to be a bit daunting, our experience is that the material usually not a problem for the students. Note that many of the topic areas in this unit can be motivated via spreadsheet and simulation examples.

3.3 Hand Simulations

Before students undertake programming and modeling in an actual simulation language, the instructor can run the class through a series of hand simulations. These hand simulations explicitly show how entities flow through a process, including any entity interactions that occur. High school students are particularly at ease with simple queueing models, even with augmentations including varying service disciplines, different service priorities, and different resource rules. A number of apropos hand simulation exercises are available in the remarkable text by Chelst and Edwards (2005), much of which one can access on the website www.hsor.org/ (sponsored by INFORMS, the Institute for Operations Research and the Management Sciences).

A typical queueing example, abstracted from Goldman (2007), is given as follows.

Example 1 *A Single-Server Queue.* Here we want to simulate the behavior of a single-server queueing system. Suppose that six customers arrive at a bank at the following times, which have been generated from some appropriate probability distribution.

3 4 6 10 15 20

Upon arrival, customers queue up in front of a single teller, and are processed in a first-come-first-served manner. The service times corresponding to the arriving customers are

7 6 4 6 1 2

We assume that the bank opens at time 0 and closes its doors at time 20 (just after customer 6 arrives), serving any remaining customers.

Table 1 and Figure 1 trace the evolution of the system as time progresses. Table 1 keeps track of the times at which customers arrive, begin service, and leave. Figure 1 graphs the status of the queue as a function of time; in

particular, it graphs $L(t)$, the number of customers in the system (queue + service) at time t .

Table 1: Tracking customers in the single-server queueing system of Example 1.

customer i	arrival time A_i	begin service B_i	service time S_i	depart time D_i	wait W_i
1	3	3	7	10	0
2	4	10	6	16	6
3	6	16	4	20	10
4	10	20	6	26	10
5	15	26	1	27	11
6	20	27	2	29	7

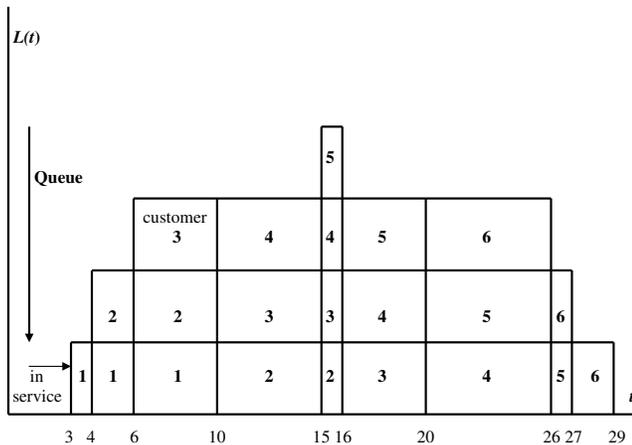


Figure 1: Status of the single-server queueing system of Example 1

Students have no trouble discovering that customer i can begin service only at time $\max(A_i, D_{i-1})$, that is, the maximum of his arrival time and the previous customer's departure time. The table and figure are then quite easy to interpret. For instance, the system is empty until time 3, when customer 1 arrives. At time 4, customer 2 arrives, but must wait in line until customer 1 finishes service at time 10. From the table, the students can see that average waiting time for the six customers is $\sum_{i=1}^6 W_i/6 = 44/6$. Further, slightly more-advanced students can calculate the average number of customers in the system as $\int_0^{29} L(t) dt/29 = 70/29$; or we could have computed the integral by adding up the rectangles in Figure 1. \square

3.4 Spreadsheet Simulation

Students can learn the basics of Excel fairly quickly, and can then program small Monte Carlo simulations in either Excel or add-in packages such as @Risk or Crystal Ball

— though it will take teachers some additional time to go over those software tools. Of course, the teacher can avoid the spreadsheet learning curve by presenting students with canned, ready-to-go applications that do not involve any programming. Examples of spreadsheet simulations include financial portfolios, certain sports pools, lottery simulations, and other interesting activities.

3.5 Random Variate Generation

In this module, the students learn about generating randomness, first with simple mechanical experiments such as coin and dice tosses, then with ancient random number tables, and then via Excel and other software programs such as Arena. As in an undergraduate course, we start with uniform random number generation, and then proceed to standard probability distributions such as the exponential and normal. We can avoid any messy mathematics, and graphically illustrate that the usual algorithms do the job.

3.6 Input Modeling

We can enhance students appreciation of input analysis by using pre-packaged demonstrations to illustrate the pitfalls of improper input modeling. For example, what if we assume that arrivals are constant when in fact they are Poisson? In addition, we have the students participate in hands-on data collection activities. For instance, we typically organize a trip to a local fast-food restaurant to collect arrival-time and service-time data. Although this is certainly an entertaining activity, the students will quickly see that it is a difficult one — but that is exactly the point.

3.7 Output Analysis

Armed with a working simulation, the next step is to perform output analysis to answer whatever questions of interest are pertinent to the problem at hand. Students will analyze descriptive data, and will even use techniques for calculating appropriate confidence intervals — though this task is a little more difficult than one might realize, since simulation output is rarely independent, stationary, or normal. Students will also investigate sensitivity issues — what happens to the output when we tweak input parameters such as arrival rates or service rates? Instructors can use output from, say, a simple single-server queueing system to demonstrate widely used output analysis techniques such as independent replications and batch means.

3.8 A Discrete-Event Simulation Language

Perhaps the highlight of any simulation course is the material on simulation modeling that uses a discrete-event language — material that can be taught at any time during the course,

possibly at the beginning and sprinkled through the rest of the course. It has been our experience that even middle school students can learn the basics of a graphical process-interaction simulation language such as Arena, ProModel, or Extend (to name a few).

The trick is to keep everything simple: Customers arrive at a facility, attempt to use resources, wait in lines if necessary, undergo rework or additional routing if necessary, and then leave — all while competing with other customers for resources. Student versions of most of the popular discrete-event languages are essentially free (or close to it), and are almost regarded as exciting new types of video games by the kids. We have *never* had any problems getting the students up to speed on the basics of the language under study, and after just one or two days, the students are accomplished enough to be able to write some fairly impressive programs.

4 DOES IT WORK?

Over the last ten years we have taught numerous short (1-day) and full (two-week) versions of the course to high school students and their teachers in various lectures, workshops and camps throughout the country. In general, the courses have been met with great enthusiasm, and student/teacher feedback is very positive. In fact, we hear from former students and teachers frequently, and the anecdotal evidence indicates that the courses have excited the students and have enhanced their appreciation of mathematics. We have not conducted any rigorous comparison trials, but those are in the offing.

5 WHERE TO GO FROM HERE

All of our experiences with simulation talks, short courses, and full courses for high school students/teachers have been terrific. Such courses have taken place as part of INFORMS conferences, at local schools throughout the country, at high school teachers conferences throughout the country, and at government-sponsored mathematics camps for kids.

What needs to be done is to continue to promote operations research in general and simulation in particular by volunteering to give additional talks, workshops, and mini-courses. At the same time, recent initiatives in states such as Michigan, North Carolina, and Georgia bode well for the formal inclusion of this type of course in the statewide high school curriculum.

ACKNOWLEDGMENTS

This material was supported by the National Science Foundation under Grant Number DMI-0400260. I am grateful for the help and encouragement of my colleagues Ken Chelst (Wayne State), Tom Edwards (Wayne State) Don Gross

(George Mason), Donna Llewellyn (Georgia Tech's Center for the Enhancement of Teaching and Learning), Matt Rosenshine (Penn State), Laurel Travis (Virginia Tech), Col. Frank Trippi (U.S. Army, Ret.), and Donna Whiting (Georgia Tech CEISMC).

REFERENCES

- Banks, J., J. S. Carson, B. L. Nelson, and D. M. Nicol. 2005. *Discrete-Event System Simulation*, 4th edition, Upper Saddle River, New Jersey: Prentice-Hall.
- Chelst, K. R., and T. G. Edwards. 2005. *Does This Line Ever Move? Everyday Applications of Operations Research*. Emeryville, California: Key Curriculum Press.
- Goldsman, D. 2007. Introduction to simulation. In *Proceedings of the 2007 Winter Simulation Conference*, S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, eds. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

AUTHOR BIOGRAPHY

DAVID GOLDSMAN is a Professor in the School of ISyE at Georgia Tech. His research interests include simulation output analysis and ranking and selection. He is an active participant in the Winter Simulation Conference, having been Program Chair in 1995, and having served on the WSC Board of Directors since 2002. His e-mail address is sman@gatech.edu, and his web page is www.isye.gatech.edu/~sman/.