Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

TECHNOLOGY TRANSFER OF SIMULATION ANALYSIS METHODOLOGY: ONE MAN'S OPINION

Barry L. Nelson

Department of Industrial Engineering & Management Sciences
Northwestern University
Evanston, IL 60208 USA

ABSTRACT

In this tutorial I provide some thoughts based on my own experience about how the analysis methodology research community might be more successful in having a meaningful impact outside of our group. Areas discussed include why technology transfer has seemed to be more effective in optimization and statistics; the audience for analysis methodology; various modes of technology transfer; technology transfer specifically to commercial software; educating consumers of analysis methodology; and how we might think differently about research.

1 INTRODUCTION

The research community in general, and the Winter Simulation Conference in particular, has a rich history of fundamental advances in simulation analysis methodology, with the promise of more to come. See, for instance, Nelson (2004) for the past, and Nelson (2016) for the future. The progression from researchers publishing ad hoc truncation and batching algorithms when I was a graduate student to the level of rigor expected of even routine contributions today is astounding.

Unfortunately, it is hard to argue that we have had as significant an impact outside of our research community as inside; that is, beyond of us talking to each other. Partly this is the nature of research: there will always be more research done than makes it into practice because it is hard to anticipate what results will end up being valuable. Research is an investment and you need to diversify if you are going to find the breakthroughs. Nevertheless, I think we have under achieved, and not because what we do has no value to simulation practice. I could, and have in various venues, illustrate all sorts of costly mistakes that can occur from poor simulation design and analysis; my focus here is on why we have not helped the situation more.

Based largely on my own experience, this tutorial will provide some thoughts about what is often called "technology transfer," but what I think of as having impact outside of our immediate research community. *These are opinions.* I have not studied technology transfer in general, but I have had some limited success with my own research and have been a keen observer of both the simulation and statistics research and practice communities. I have been a practitioner as well as a researcher. Research that I have done that has seen the light of day includes ranking & selection procedures (in particular screening (Nelson et al. 2001) and KN (Kim and Nelson 2001)); simulation optimization (Industrial Strength COMPASS (Xu et al. 2010) and the concept of "clean up" (Boesel et al. 2003)); a graphical display (MORE plot (Nelson 2008)); multivariate input modeling (NORTA and its derivatives (Biller and Nelson 2005)); and output analysis for input sensitivity ("input uncertainty" (Song and Nelson 2015)). Some of these are in commercial software, some were made available by my students and me via the web, and some were implemented by others.

My focus here is dynamic, stochastic simulation that could be discrete event or agent based, but the main ideas probably apply to deterministic computer experiments as well. By "analysis methodology" (AM) I will mean input modeling, experiment design and output analysis; see, for instance, Nelson (2013).

2 TWO BENCHMARKS

Almost by definition, "computer simulation" requires software, which makes us a software-oriented field. Anything we do in AM that is applied has to be committed to software.

Two of the most closely related research communities to simulation are statistics and optimization. Neither a modern statistical analysis nor a large-scale optimization can be accomplished without software. Both communities have researchers pushing the cutting edge of theory, and both have open-source, free and commercial products for practitioners and other researchers to use. Superficially this seems just like us, but there are differences.

In statistical software it is a competitive advantage to have implemented the latest in *useful* research results. On the commercial side the vendors employ Ph.D.-level researchers whose job, at least in part, is to evaluate new research for inclusion in their products. On the open-source and free side (such as `R`) researchers are contributing their methods to software archives, or the latest methods are being coded by others and contributed. Other than experiment design and cleaning, statistical software is not much concerned with the *creation* of data, but is very concerned with getting the most reliable insight out of the data that are available. *The competitive advantage comes from providing more robust and deeper insight than can be provided by others.*

Statistical software has to attack a wide variety of data analysis problems. Optimization software attacks one problem: the mathematical program. I realize that this is an over-simplification, but the near unanimous agreement on the form of a "mathematical programming model" is important to stress. *The existence of a standard allows the model/problem to be separated from the solver that will be used to attack it.* Thus, the competition is on providing better and better solvers, and not as much on better modeling environments (although, of course, the interface does matter). "Better" means faster or solutions closer to optimality.

Other than random-number and random-variate generation, I would argue that technology transfer in statistics and optimization has been significantly more successful than in simulation AM, and two reasons stand out:

1. In simulation, competition has been on *generating the right data*, not on *the best analysis of it*, while in statistics analysis of the data is pretty much the whole game.
2. In simulation, "generating the right data" has focused on modeling environments, which means that it is a competitive advantage to *differentiate based on a better modeling environment*, not on *better/faster solution of a common model* as in optimization.

These two differences have had a profound impact on the acceptance of, or frankly even a sense of the need for, better AM. This implies, to me, that for us to be successful in technology transfer we have to do several things well that maybe statistics and optimization do not:

- We have to understand simulation users and how they use simulation (as opposed to how we think they should use simulation).
- We have to see the value proposition of our methods from the user's point of view, not our point of view.
- Related to the previous point, we need to understand *general-purpose* vs. *special-purpose* use. There is a place for both, but they are not the same place, and time consumed plays a big role here.
- We have to make outcomes and benefits interpretable, and methods implementable; this may require *compromising* our results in certain ways.

In the remainder of this tutorial I will expound on some of these points.

Remark: It is interesting to speculate as to why we were and are so much more successful in technology transfer for random-number and random-variate generation than in other areas of AM. The golden age of random-variate generation coincided with very constrained computing capabilities. Therefore, speed really did matter, and many of the research advances were in the form of faster random-variate generation algorithms. There also seems to be an understanding at all levels (practitioner, vendor, consultant and researcher) that it all starts with the random numbers, and if they are defective the rest does not matter. This may have resulted from well-known spectacular failures like `RANDU`. Sometimes a disaster is the best advertisement for better methods.

3 AUDIENCES FOR AM TECHNOLOGY TRANSFER

To whom might we transfer our technology? There is more than one audience.

Commercial Software Vendors: This is the quickest way to reach a wide audience, since much, if not most, simulation practice uses commercial software. In this environment the key is to be *general purpose*. For good reason, the software vendors are only interested in investing the effort to implement AM if it will be considered useful by many of their users. A tool that appeals to a narrow audience, or that requires a very sophisticated user, seems not to be worth the effort, at least in their current user environment. *Keep in mind that simulation software is not like Excel: the market for it is relatively small, and therefore the software is relatively expensive.* In fact, the market is not as large as that for statistics or optimization software, either, and the size of the market has an influence on investment in development and coverage of more niche features.

Open-Source/Free Software: There is open-source, or at least freely available, simulation software; see, for instance, SimPy (`simpy.readthedocs.org`). To the best of my knowledge the focus is still on modeling, not AM. Model development in most, but not all, of this software is more demanding of the user than commercial software (but see, for instance, JaamSim `jaamsim.com/`). However, for us the implementation of new AM tools would be easier. If something like SimPy ever dominated simulation modeling, then AM would be in business because we could code our methods in Python and contribute to public Python libraries. I am not holding my breath.

Individuals and Companies: This is probably where we have had our greatest undocumented successes. Certain industries, for instance security pricing and risk analysis in finance, are consumers of *special-purpose* simulation methodology because they need precise answers to computationally difficult problems in a timely manner. Their needs more closely align with our standards for publishable research than the needs of commercial software vendors. Pushing the boundaries matters.

Other Researchers: We build on each other's work, which means we often find ourselves implementing each other's stuff because we do not have a common language for distributing code. Frankly, we do not put much emphasis on getting our code to a level at which we would be willing to share it (there are notable exceptions, including `simopt.org/`). We also touch researchers in other quantitative fields who exploit our ideas to facilitate their own research; e.g., queueing approximations are often tested using simulation, and the spread of disease is often simulated. Researchers in other fields sometimes see the need for, or motivate creation of, state-of-the-art AM.

4 MODES OF TECHNOLOGY TRANSFER

What modes do we have, and how effective are they?

Books: Law and Kelton (1982), and its many follow-on editions, showed just how influential books can be in transferring AM technology. What made it so successful? First and foremost it was readable, striking just the right balance between intuition and being technically precise. Specific methods and algorithms were provided, as well as the basic open-source simulation software `simlib`. But in a stroke of genius, the presentation of the AM tools did not depend on, nor were they even presented in the context of, `simlib`. Providing software, while at the same time being language independent, was a strategic choice that I believe paid substantial dividends. That said, in 1982 the ease-of-use gap between `simlib` and the commercial software was not that large, which is not true today. I doubt that a language-independent book can have the reach of Law and Kelton (1982) now, and once you start mixing software instruction with AM instruction, the AM part seems to lose out. Nevertheless, books are far more effective at reaching reseachers in other fields than journal papers are: when I wanted to learn something about Gaussian Markov Random Fields for my research, I started with books not papers.

Web pages: If you put your code up on your web page, people find it. Purchasing domain names (e.g., `www.iscompass.net`) is even more effective. This approach most often reaches sophisticated individuals and other researchers who know what they are looking for; nevertheless, it is an outlet that we did not have in 1982 and we should exploit it.

Commercial Software: If it is in the commercial software, and promoted by the vendor, then it will be used. Period.

Consulting: My best undocumented technology transfer has occurred in consulting projects. Most of us would benefit by undertaking some consulting: it gives you a new perspective on simulation users and uses, and you can often provide the client with a competitive advantage by employing good AM. Of course there are professional consultants who use simulation, and I suspect that if we developed research reports/videos/handbooks *designed specifically for such consultants* we would make a significant leap in practical impact.

Journals and Conferences: These are mostly for communicating with each other. Thank goodness for WSC: it gets new results out quickly, in written and reviewed form, and is easily accessible to everyone via the web. WSC papers are among my most cited. Enough said.

5 TECHNOLOGY TRANSFER TO SOFTWARE

One of my research goals is to create AM tools that are appropriate for, and are adopted by, commercial software vendors and simulation users, and I have had a few successes. Here are some thoughts about what it takes.

5.1 Understanding Simulation Applications

Many of us do not have a clear vision about how simulation software is used in practice. Without this sense, it is hard to create AM tools that simulation users/vendors want.

The sequence *define problem → collect data → build model → validate model → run experiment → make decision* rarely happens. The problem definition almost always evolves as the simulation provides insight. Data collection is rare; finding data from existing sources, ranging from large, unstructured data archives to people who know the process is more typical. Debugging and experimentation occur together, and validation (sadly) often goes no deeper than "do the results make sense?" It is within this environment that AM tools have to work.

Because simulations are being run during almost every phase of the project, the idea of one big, well-defined simulation experiment is a myth. I have come to realize that this is why the simulation software vendors are worried about AM methods that take too much time either to implement or to execute: they know users will be running the methods over and over again and will be impatient if each experiment takes a long time. This is unfortunate, but it is reality. Also, simulation results are almost always inputs to a decision; they do not make the decision. This is why providing sets of good solutions, trade-offs, sensitivities, and measures of risk are important. Single answers are rarely relevant.

5.2 Understanding Simulation Software Users

No doubt it is dangerous to try to make general statements about simulation software users. But I think it is safe to say that unless the users are simulation consultants, they are not typically specialists in simulation, and sometimes see simulation as little more than dropping the correct icons on the workspace. To me this means that they most often need tools that are general purpose—so it is not too difficult to decide what to use—that give results that are easy to interpret correctly, and that have robust defaults for any settable parameters. Of course, we also need to remember that simulation software users operate in the context described above. *They would like to feel safe in using methods without having to fully understand them, and therefore they depend on the software vendor to only give them well-accepted, trusted methods.* This likely makes commercial software vendors cautious about adopting new AM.

5.3 Commercial Software

It is easiest (not to say easy) to reach the commercial software with AM tools that sit entirely on the front end of the simulation (e.g., input modeling), or entirely on the back end (e.g., the MORE plot of output) of the simulation. Methods that affect or interact with the internals of the simulation itself require a greater commitment by the vendor. I admire the strategy of OptTek Systems, Inc.: they did the work required to customize their simulation optimization engine OptQuest (`www.opttek.com/OptQuest`) for any simulation software product that was interested in having it. Because they were willing to figure out how to touch the simulation internals, they were able to implement powerful methods.

I believe there is an unintended downside of thinking that simulation software has to be more than just the modeling environment, and instead also has to include the simulation engine, experiment design, output analysis capability and sometimes input modeling. There is little doubt that this is what users want, but few vendors have the resources to excel in all of these domains.

Consider, for instance, ExpertFit (`www.averill-law.com/distribution-fitting/`) which is stand-alone input modeling software that exports fitted distributions in the syntax of a number of simulation languages. As with OptQuest, this is a lot of work, and a moving target. But if our field had chosen to have a common input, experiment design, and output data model, we would be closer to the optimization paradigm and might see many vendors competing in stand-alone input modeling, experiment design and output analysis tools. This would be good for AM technology transfer because offering the best/most robust tool would be an advantage. Imagine a software ad that touted "Want to have the most accurate assessment of your company's risk? Analyze your simulations with OAGenius, which implements the latest methods from Dr. Shane Henderson." But as long as the simulation software is comprehensive the market for stand-alone support tools will be small.

Common standards seem unlikely to happen. However, in all commercial simulation products it has become easier to import input data and export output data. Further, some products are facilitating add-ins, exposing control structures, or providing wrappers around their simulation environment. These are opportunities for us if we can convince users that there is value added, such as avoiding poor decisions that expose the company to costly risk or to underachieving desired objectives.

6 TECHNOLOGY TRANSFER IN THE CLASSROOM

The application context described in Section 5.1 should affect our teaching. This includes teaching students to report more than one recommended option; emphasizing the need for more and more careful simulation near the end of the study; and showing them how to build and explore parametric models when they do not have hard data, particularly early in a simulation project (see, for instance, Tongarlak et al. (2010)).

For those of us who teach undergraduate engineers, business students, operations researchers and statisticians their first course in stochastic simulation, we should adopt a goal of producing students who are good AM consumers. *This does not mean teaching them lots of methods.* Instead, we should develop understanding and intuition about why simulation design and analysis matters, and what can go wrong if you are not careful. Methods will evolve, but the core issues will remain.

Here are my top seven topics that, if fully understood by students, will make them care about AM:

1. The difference between risk and error.
2. The difference between statistical dependence, functional dependence and non-stationarity.

3. The possibility of sensitivity and insensitivity to input models.
4. The difference between time-dependent and time-independent performance.
5. That simulation optimization is hard, and why.
6. That a system with good mean performance may still have bad behavior.
7. That bad decisions can result from the insight provided by poor analysis (note the emphasis on the *decision*).

7 THOUGHTS ABOUT RESEARCH

I have had National Science Foundation (NSF) grants with simulation industry support, NSF Grant Opportunities for Academic Liaison with Industry (GOALI) grants with simulation industry co-PIs, and industry-sponsored simulation research projects. These help keep my research relevant and some technology transfer is almost a given.

What sort of research should we be doing? I do not want to go back to the days of ad hoc truncation rules and batching algorithms; we have advanced, and that is a good thing. But we should also understand that there are limits to limits. At the risk of losing all of my friends, let me belabor this point:

Knowing that, in steady-state simulation, the Bias^2 of the estimator $\bar{Y}$ is often $O(1/m^2)$ while the variance is $O(1/m)$, where m is the run length, was a profound insight that helped us know where to put our emphasis in the days of single-processor computers. But knowing that some estimator $\widehat{\theta}$ satisfies a Central Limit Theorem involving unknown parameters that are difficult or impossible to estimate is much less helpful for applications. I appreciate the benefit of limit theorems, but we should not elevate them to being the *only* measure of value in AM research. We (rightly) lose credibility with users and vendors when we do. *The most practically important limit theorems are the ones that affect how or whether you implement a method.*

We could provide much better value propositions for our methods if we established their robustness, or at least that they degrade gracefully when unverifiable assumptions are not satisfied. Better yet would be detecting and alerting when our methods are *not* working. This is hard, but compelling.

We should always push the boundaries of theory: this is a critical investment. But perhaps we should consider a paper to have *additional value* if it also includes an out-of-the-box implementation with carefully considered and studied compromises. This might close a little of the gap between research and impact.

8 CONCLUSIONS

From my perspective, AM research is thriving and working on interesting problems. The diminished barrier to using parallel computing has opened new frontiers for simulation design and analysis. In the long run, however, relevance matters as much as truth and beauty, and it is within our power to become more relevant.

ACKNOWLEDGEMENTS

This research was partially supported by the National Science Foundation under Grant Number CMMI-1537060 and GOALI co-sponsor SAS Institute. All opinions are those of the author who benefitted from feedback from Shane Henderson, Emily Lada and Eunhye Song.

REFERENCES

Biller, B., and B. L. Nelson. 2005. “Fitting time-series input processes for simulation”. *Operations Research* 53 (3): 549–559.

Boesel, J., B. L. Nelson, and S.-H. Kim. 2003. “Using ranking and selection to clean up after simulation optimization”. *Operations Research* 51 (5): 814–825.

Kim, S.-H., and B. L. Nelson. 2001. “A fully sequential procedure for indifference-zone selection in simulation”. *ACM Transactions on Modeling and Computer Simulation* 11 (3): 251–273.

Law, A. M., and W. D. Kelton. 1982. *Simulation modeling and analysis*. McGraw-Hill Book Company.

Nelson, B. 2013. *Foundations and methods of stochastic simulation: A first course*. Springer Science & Business Media.

Nelson, B. L. 2004. "50th anniversary article: stochastic simulation research in Management Science". *Management Science* 50 (7): 855–868.

Nelson, B. L. 2008. "The MORE plot: Displaying measures of risk & error from simulation output". In *Proceedings of the 2008 Winter Simulation Conference*, 413–416. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Nelson, B. L. 2016. "Some tactical problems in digital simulation for the next 10 years". *Journal of Simulation* 10 (1): 2–11.

Nelson, B. L., J. Swann, D. Goldsman, and W. Song. 2001. "Simple procedures for selecting the best simulated system when the number of alternatives is large". *Operations Research* 49 (6): 950–963.

Song, E., and B. L. Nelson. 2015. "Quickly assessing contributions to input uncertainty". *IIE Transactions* 47 (9): 893–909.

Tongarlak, M. H., B. Ankenman, B. L. Nelson, L. Borne, and K. Wolfe. 2010. "Using simulation early in the design of a fuel injector production line". *Interfaces* 40 (2): 105–117.

Xu, J., B. L. Nelson, and J. Hong. 2010. "Industrial strength COMPASS: A comprehensive algorithm and software for optimization via simulation". *ACM Transactions on Modeling and Computer Simulation* 20 (1): 3.

AUTHOR BIOGRAPHY

BARRY L. NELSON is the Walter P. Murphy Professor in the Department of Industrial Engineering and Management Sciences at Northwestern University and a Distinguished Visiting Scholar in the Lancaster University Management School. He is a Fellow of INFORMS and IIE. His research centers on the design and analysis of computer simulation experiments on models of stochastic systems, and he is the author of *Foundations and Methods of Stochastic Simulation: A First Course*, from Springer. His e-mail address is nelsonb@northwestern.edu.