SAND2019-5640C

Design of Experiments for Cyber Experimentation

Laura P. Swiler, *Optimization and Uncertainty Quantification Dept., Sandia National Laboratories, Albuquerque NM 87185-1318*
Michael G. Stickland, Thomas D. Tarman, *Experimental Cyber Initiatives Dept., Sandia National Laboratories, Albuquerque NM 87185-3696*

Abstract

As cyber researchers increasingly rely on virtualized testbeds, there is a need to use formal statistical methods to generate ensembles of cyber experimental runs or emulations. Principled statistical approaches can provide rigor in analyzing results and support inferences made from such results, for example about the importance of certain parameters or statements about the range or percentiles of likely responses. In this paper, we outline terminology used in the Design of Experiments (DoE) community, provide background about DoE methods used for computational simulations such as engineering codes, demonstrate the use of formal DoE methods on a cyber experimental testbed, and develop recommendations for future work in DoE within the cyber experimentation community.

1. Introduction

There are many issues associated with defining realizations of cyber experiments on testbeds. Several frameworks have been developed to make the experimental design construction more consistent and reproducible. For example, DEW (Distributed Experiment Workflows) [10] provides a generic descriptive language that encodes the desired behavior of an experiment. DEW provides a way to describe a scenario and associated topologic constraints, with the focus of developing more automated testing scripts, automated generation of scenarios and topologies, and sharing and reuse. As another example, Maricq et al.[9] examine platform variations in emulation-based experiments using carefully structured experiments and statistical analysis. They present many good practices, such as the need to randomize experimental runs, eliminate consistent outliers, and being cognizant of shared resources where other workloads may significantly affect the emulation results.

In this paper, we assume the reader has access to a cyber testbed and wants to generate multiple scenarios or realizations for the purposes of sensitivity analysis, uncertainty quantification, or optimization. There is a rich history of performing such ensemble studies in the computational science community. [11,13] For example, researchers developing and running structural dynamics, shock physics, or heat transfer simulations typically run ensemble simulations to understand safety critical scenarios. Accurately assessing the sensitivity of the results with respect to important parameters and assessing uncertainty in the responses (e.g. what is the probability that the displacement at a certain location is greater than X?) are critical for assessing the reliability of systems in the automotive and aerospace industry, for example. [1,2,12].

We draw on the history of experimentation on computational simulations in this paper with the goal of providing guidance on how one specifies the parameters to be investigated (e.g. with discrete levels or settings, as uncertain parameters with continuous or discrete distributions, etc.), how one generates the sample set, and how one analyzes the results.

Currently the state of the art is to either do a "hero" calculation on a cyber testbed with an experiment that is run once or just a few times, or to perform a "grid" study where the parameter settings are discretized and each combination of parameter settings is run. This latter can be computationally expensive due to the curse of dimensionality and thus methods requiring one to run only a subset of the complete enumeration of the parameter space are necessary. We discuss grid studies (e.g. full factorial designs) but also suggest other options in instances where the number of runs is much less than the number of parameter combinations that is possible. This reduces computational cost at the expense of understanding interaction effects between variables. Finally, some cyber experiments are run multiple times at the same settings (sometimes called replications or iterations) to understand the stochastic behavior of the system. The optimal number of replications is also a topic of interest and this is a topic little discussed in the cyber experimentation community although there is some recent work demonstrating number of samples required to achieve a particular confidence interval for the mean or median of a set of runs [9].

We feel that terminology is a very important and thus Section 2 provides detailed definitions of terms related to Design of Experiments. Section 3 provides an example demonstrating DoE on a cyber problem, and Section 4 provides recommendations.

2. Definitions

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

We provide definitions of terms in Sections 2.1-2.10. Sections 2.11 and 2.12 address topics which are often a source of confusion and for which there is active debate.

2.1. Experimental Design

This is an overloaded term. It can be used to refer to how one selects input parameter settings for ***one*** experiment (e.g. which virtual environment to use, how many virtual machines, what protocols to run, etc.) The one experiment may involve multiple replications where the experiment is run multiple times at the same settings of the governing parameters.

However, Experimental Design can also be used to refer to the broader problem of selecting a set or suite of experimental parameter settings at which one will run the cyber experimental model (e.g. various choices of number of cores per machine, protocols and environment settings, packet size of traffic, bandwidth of links, etc.)

2.2 Design of Experiments

To make the distinction between Experimental Design and Design of Experiments clear, we refer to Experimental Design as the process of selecting parameter choices for one run (with or without replicates) and the process of selecting an ensemble of runs as Design of Experiments since the reference to "Experiments" in "Design of Experiments" makes clear one is generating an ensemble.

The goal of DoE is to generate an "ensemble" of runs, where each run itself may involve replicate runs or not. The parameter values for each run should be carefully chosen to extract as much trend data from a parameter space as possible using a limited number of sample points. Additionally, the ensemble can be used to perform sensitivity analysis and uncertainty quantification studies. For example, one might want to know which variables contribute the most to packet response time or determine the distribution of quantities of interest such as network latency, bandwidth, and memory.

2.3 Optimal Experimental Design

The selection of input parameters for design of experiments may be done in many ways. There are several criteria one can choose to optimize when selecting a design. For example, Monte Carlo sampling over the parameter domain generally tries to select points with good "space filling" properties. There are several "alphabet-optimal" designs such as A-optimal, B-optimal, D-optimal, G-optimal, and I-optimal. These designs all involve optimizing some property of the run matrix.

Let us denote the run matrix, X, to be of size nxp, where the n rows represent n runs, each with p parameter values. The information matrix is denoted by the inverse of the variance matrix, or $[X^T X]^{-1}$. An A-optimal design minimizes the trace of the inverse of the information matrix which results in minimizing the average variance of the estimates of regression coefficients built on the dataset X. A D-optimal design minimizes the determinant of the information matrix which results in maximizing the information content of the parameter values (this is also has the effect of good "space filling" properties). A G-optimal design minimizes the maximum variance of the predicted values from a regression fit built on the dataset X, etc. [18]

The above designs are fixed designs that seek to optimize a particular property. There are also adaptive designs in which an initial experiment is run and then an optimization procedure identifies the "next best" experiment to run to optimize some objective. Typically, the objective involves improving the parameters of the model and "gaining the most information" possible. For example, Bayesian optimal experimental design has become popular, with the goal of determining experiments which most inform the posterior distribution inferred on model parameter values (need citation).

2.4 Uncertainty Quantification

Uncertainty Quantification (UQ) is the process of characterizing all uncertainties that could affect the results of the cyber experimental runs. Once the uncertainties are identified and characterized as "input uncertainties", they are propagated (e.g. mapped) through the experiment to obtain uncertainties on the results ("output uncertainties").

UQ is a closely related activity to V&V and essential for verifying and validating computational models. The goal of UQ is to propagate input distribution uncertainty through the model to generate distributions on the model responses. This can then be used to understand the mean and variance of the output, calculate the probability that the response is less than or greater than a particular threshold value, etc. UQ, along with V&V, enables modelers and analysts to make statements about the degree of confidence they have in their simulation or emulation-based predictions. Uncertainty quantification has been a fundamental capability supporting nuclear reactor safety studies, performance assessment of repositories for the disposal of nuclear waste, computational fluid dynamics for aircraft design, and climate model predictions [3,7,11,12]. We anticipate more widespread use of UQ in the cyber emulation community to address questions about the performance and confidence in mitigation strategies for network attacks, for example. However, emulated cyber environments are different from physics simulation models used in many risk assessments of engineered systems. We need to understand how typical UQ methods work in the presence of stochastic network behavior, and how to use UQ methods to identify "edge case" behavior where software, hardware, network topology, and vulnerabilities interact in unforeseen ways.

2.5 Sensitivity analysis

Sensitivity analysis (SA) is the process of identifying the most significant factors or variables affecting the uncertainty of the Emulytics model predictions [16,17]. This can help identify where to most effectively place cyber threat mitigations or invest in resources. Sensitivity analysis can be used to identify model inputs in which a reduction of uncertainty would most reduce the uncertainty of the model output, or to identify model inputs that could be fixed to simplify the calculation, or to identify general trends between inputs and outputs. Sensitivity analysis (SA) can be performed using local or global methods.

2.6 Verification and Validation (V&V). Over the past few decades, the computational simulation community has developed a strong emphasis on Verification and Validation activities to build credibility in scientific computing. A study by the National Research Council at the National Academies issued a report outlining the mathematical and statistical foundations of V&V and UQ as primary activities supporting the reliability of computational models [12]. A number of professional societies have developed guidelines and standards for V&V activities [1,2]. We take as definitions those outlined in [13]:

• *Verification* is the process of assessing software correctness and numerical accuracy of the solution to a given mathematical model.

• *Validation* is the process of assessing the physical accuracy of a mathematical model based on comparisons between computational results and experimental data.

Verification provides evidence that the model and the equations are correctly solved. In computational simulations, it deals with the adequacy of the numerical algorithms to provide accurate numerical solutions to the discretized partial differential equations. In cyber experimentation, it can refer to how accurately the virtualized software and hardware components represent their physical counterparts. Validation addresses a different question: the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model. Validation provides evidence that the cyber experiment is appropriate for the problem of interest. Validation typically involves measuring agreement between the experimental outcomes and "gold standard" outcomes from appropriately designed validation experiments running on actual networks or physical testbeds with no emulation. The extent to which validation can be performed on cyber experiment models and how to do it is an open research question [8].

2.7 Parameter Study

Typically, a parameter study means the same thing as an experimental design: it specifies a number of runs which involve varying the allowable levels of the parameters in a structured way.

2.8 Factorial Design

A factorial design is an experimental design that samples the full combination of all parameters. Thus, if there were 3 parameters and each had 5 allowable values or levels, a full factorial design would involve 5*5*5 = 125 runs. A fractional factorial design only involves a subset of the full factorial. The subset is typically chosen to best estimate the main effects of the parameter values.[6] There is a rich statistical literature based on orthogonal arrays that involves determining fractional designs. The approaches typically involve substantial computation, rely on libraries of pre-generated orthogonal arrays, and are mainly valid for combinations of variables only with two or three levels.

2.9 Replicates

A replicate refers to running the same set of experimental settings multiple times to see how the response varies within that setting. A replicate can also be called an iterate. The idea of replicates comes from the early experimental design literature. A common example is that of crop yields, where the parameter of interesting might be the application of fertilizer. A "replicate" would be one of several plots to which fertilizer was applied or one of several plots to which it was not applied. Replication is needed when there can be significant variation within a treatment or combination of parameter settings.

2.10 Surrogates

Experimental design and UQ can both require huge numbers of model evaluations to generate accurate statistics or to perform sensitivity analysis. For this reason, the computational science community has embraced the notion of "surrogate models", also called emulators, meta-models, or response surface approximations. Surrogate models are used in computational models for physics and engineering applications to replace the "full physics code runs" which involve the solution of partial differential equations over very large (e.g. > 1M elements) meshes. In the past two decades, surrogate modeling for computational science problems has become an active research field. Some of the most common surrogate models involve regression [20], Gaussian processes [14], and polynomial chaos expansions [5,21].

Cyber testbeds themselves may be considered surrogates for real-world environments. However, it is also possible to think of surrogates or lower-fidelity models for cyber virtualized experiments or emulations. For example, such surrogates could be regression models or other statistical data-fit models such as Gaussian processes. But surrogates for cyber experiments might also involve discrete event simulators such as the NS3 network traffic simulator. Finally, surrogates

for cyber experiments might involve analytic formulas. One use of surrogates for cyber experiments is for "multi-fidelity" UQ. In this approach to UQ, a low-fidelity model is run thousands of times, where a high-fidelity model may be run a few times. The results are combined to produce a high-fidelity estimate which has the benefit of low variance from the large number of low-fidelity runs and improved accuracy from the high-fidelity runs which reduce bias in the estimate [4].

2.11 UQ vs. Experimental Design

Note that there can be a subtle difference in how one treats the results of an experimental design study and a UQ study. Typically, uncertainty quantification requires the user to specify probability distributions on the input parameters (e.g. normal, Weibull, exponential, etc.). Then, samples are taken according to the probability distributions and the model is run at those settings to produce a distribution on results. Thus, uncertainty quantification focuses on mapping input distributions to output distributions: the goal is understanding the probability distribution of the output and associated statistics such as mean, variance, and percentiles of the output. Historically, experimental design methods do not require distributions. They are more focused on the influence of the input settings (often taken to be binary or discrete levels). Thus, the goal of experimental design is to say something like "the application of fertilizer results in a mean crop yield that is statistically significantly higher than without the fertilizer." Parameter studies, factorial studies, and parameter sweeps over levels of an input parameter typically are not focused on "distribution of inputs to distribution of outputs mapping" but instead on "what is the difference in response under various experimental settings?" or "what is the trend in the response as we increase the value of an input parameter?"

A confusing aspect of the distinction outlined above is that Monte Carlo sampling may be used for both UQ and experimental design. That is, often Monte Carlo methods are used to generate realizations of input parameters for UQ. However, Monte Carlo methods may be used to generate a small number of samples from a high dimensional space when a full factorial design or complete enumeration is too expensive. In the latter case, one does not necessarily impose a distribution structure on the outputs. It is acceptable to use Monte Carlo methods for both UQ and experimental design studies, but the analyst should carefully state what assumptions are being made on the input distributions or levels of parameter values.

2.12 DoE for physical vs. computational experiments

Statisticians classify DoE approaches into two different areas: classical Design of Experiment methods and the more modern design and analysis of computer experiments (DACE) methods. Classical DoE techniques arose from technical disciplines that assumed some randomness and nonrepeatability in field experiments (e.g., agricultural yield, experimental chemistry). DoE approaches such as central composite design, Box-Behnken design, and structured factorial designs have approaches to generate and handle replicate runs. These designs also put sample points at the extremes of the parameter space, since such designs offer more reliable trend extraction in the presence of nonrepeatability.

DACE methods are distinguished from DoE methods in that the nonrepeatability component is omitted for computer simulations which are deterministic (e.g. one set of input parameters always results in the same output. This is usually the case for the partial differential equations models used to solve physical problems). Thus, for DACE experiments, there are no replicates. In these cases, space-filling designs and Latin hypercube sampling are more commonly employed to accurately extract trend information. Quasi-Monte Carlo sampling techniques which are constructed to fill the unit hypercube with good uniformity of coverage are also used for DACE. Space filling designs are also employed when constructing surrogate models, and much of the early DACE work centered around sampling to construct Gaussian process models [15,18,20]. Note that cyber experimentation involves aspects of both DoE (e.g. possible randomness and nonrepeatability from stochastic network traffic, delays, timings, etc.) and DACE (large numbers of simulation parameters, need for good space filling designs).

3. Demonstration

In this section, we provide a simple example of how one might use DoE within a cyber experimentation workflow to generate and analyze results. For this demonstration, we utilize two software frameworks developed at Sandia National Laboratories. Dakota is a software framework for uncertainty quantification, sensitivity analysis (SA), optimization, etc. of computational simulations (https://dakota.sandia.gov). It has an extensive set of UQ/SA algorithms in a framework that allows one to generate automated parameter studies on a simulation or emulation code easily.

Firewheel is a toolset developed by Sandia National Laboratories to launch and manage virtual machines (VMs) to emulate networks. It includes a scriptable interface with Python APIs to support the experimentation lifecycle such as specifying topologies and parameters, launching applications, capturing data and running services.

A picture of the workflow coupling Dakota to Firewheel is shown in Figure 1. We used the workflow to drive an experiment shown in Figure 2. In this experiment, we analyzed a DNS amplification attack. The attacker sends small DNS queries to a server from a spoofed victim address. The DNS server then sends large responses to the victim. We analyzed two responses: the impact on the server's CPU utilization and congestion on the victim's link. The setup is shown in Figure 2. The input parameters varied for the DoE are shown

in Table 1. Note that for these particular scenarios, the attack rate was fixed at 100,000 packets per second.

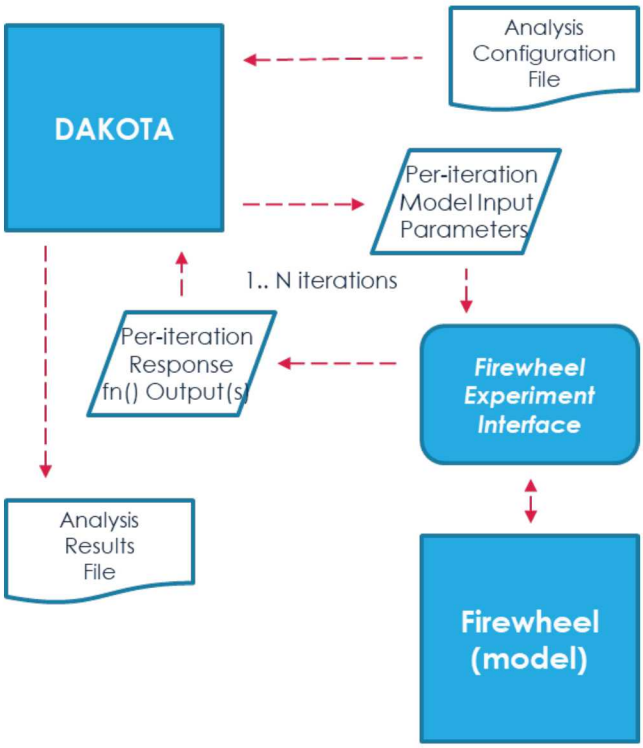

Figure 1. Firewheel Emulytics model coupled to Dakota

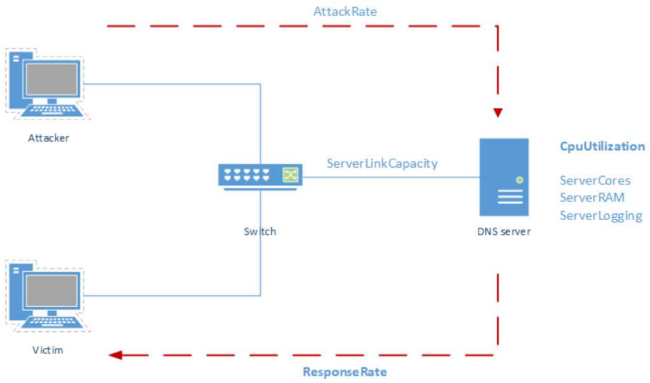

Figure 2. DNS amplification attack scenario used for purposes of illustrating Design of Experiments workflow.

Input parameter	Levels
ServerLinkCapacity	100, 1000, 10000 Mpbs
ServerCores	1,2,4,8,16
ServerRAM	256, 1024, 4096. 8192 MB
ServerLogging	True, False

Table 1: Parameters and associated levels varied in DoE.

We first performed a full factorial design, running the experiment at all combinations of the above parameter values. This DoE involves 3*5*4*2 = 120 runs. Generating this full ensemble of 120 runs took 12 hours of wallclock time.

The results are shown in Figures 3 and 4. Figure 3 shows that server logging and number of server cores (and their combination) are statistically significant for server CPU utilization. Server link capacity, however, is not. Figure 4 shows the impact of the input parameters on victim traffic (in packets per second). This figure shows that server RAM is not significant but all other configuration choices are statistically significant.

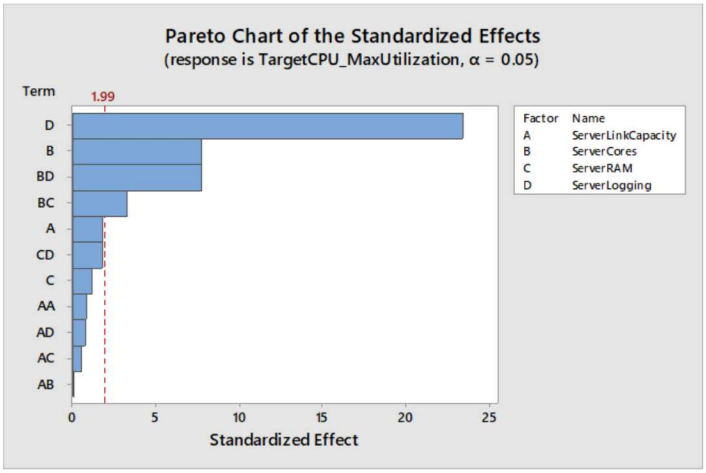

Figure 3. Ranking of important parameters for CPU utilization.

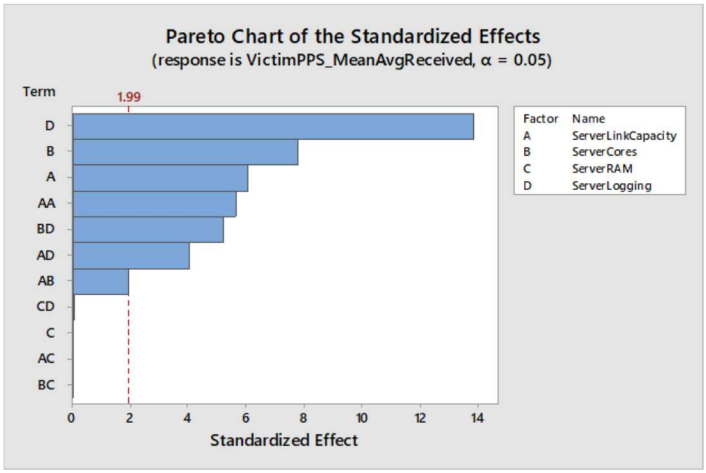

Figure 4. Ranking of important parameters for victim response rate (VictimPPS).

An important statistical analysis that can be performed on a full factorial design is called "main effects" analysis (add more description). Whereas the Pareto plots show which main and interaction effects are significant, a main effects analysis will quantitatively show the degree of effect. This information is useful when assessing the sensitivity of the output metrics on inputs, which can inform a decision maker where they need to invest in more resources to mitigate the attack.

As the main effects plots in Figures 5 and 6 demonstrate, investing in additional server memory has no effect on either metric, i.e. the server CPU utilization and the impact on the victim machine. Thus, there is no need to invest in additional memory. However, enabling DNS query logging on the server has a beneficial effect on both metrics, so the server administrator should enable it.

A more complicated situation arises when considering whether to add cores to the server. There is a strong beneficial effect on the server's CPU utilization when adding cores, but a strong detrimental effect on the intensity of the traffic received at the server. To find a configuration that simultaneously optimizes both metrics, we ran the DAKOTA/Firewheel experiment with DAKOTA configured to perform a black box, multi-objective optimization study. The results of this study are shown in Table 2. The results show a set of seven Pareto-optimal solutions which, in the absence of other criteria (e.g. cost), simultaneously minimizes victim traffic intensity and server CPU utilization.

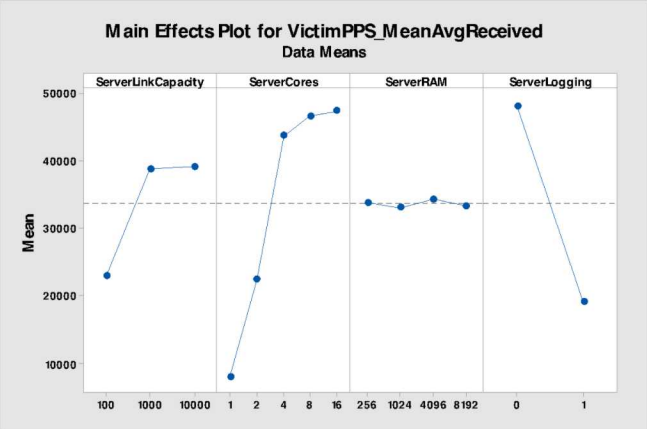

Figure 5. Main effects plots for victim attack traffic.

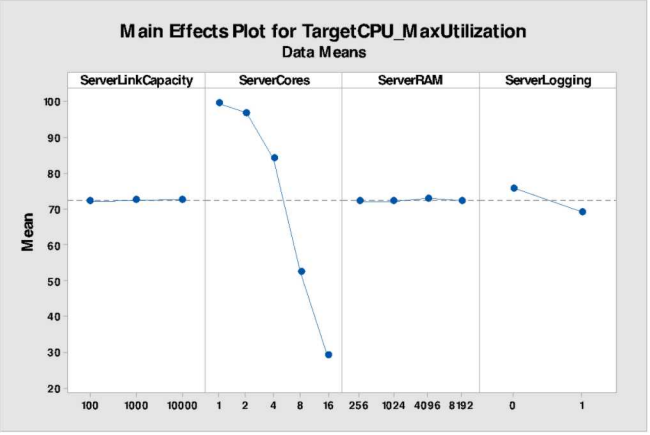

Figure 6. Main effects plots for server CPU utilization

Parameters	Results
ServerCores = 1 **ServerLinkCapacity = 100** **ServerLogging = yes**	VictimPPS = 5520 ServerCPU = 100
ServerCores = 2 **ServerLinkCapacity = 100** **ServerLogging = yes**	VictimPPS = 11559 ServerCPU = 95.2
ServerCores = 2 **ServerLinkCapacity = 1000** **ServerLogging = yes**	VictimPPS = 11615 ServerCPU = 94.9
ServerCores = 2 **ServerLinkCapacity = 10,000** **ServerLogging = yes**	VictimPPS = 11641 ServerCPU = 92.9
ServerCores = 4 **ServerLinkCapacity = 10,000** **ServerLogging = yes**	VictimPPS = 23525 ServerCPU = 79.1
ServerCores = 4 **ServerLinkCapacity = 1000** **ServerLogging = yes**	VictimPPS = 24033 ServerCPU = 78.4
ServerCores = 32 **ServerLinkCapacity = 1000** **ServerLogging = yes**	VictimPPS = 25887 ServerCPU = 12.1

Table 2: Pareto optimal solutions that simultaneously minimize victim traffic and server CPU loads

4. Future work

A future study would also consider the propagation of input uncertainties to the response metrics of interest, in order to assess model confidence and robustness, and to assist a decision maker with assessing any risks that this uncertainty may pose to the server and/or the victim. In this example, the input uncertainty would be the attack rate (in DNS queries per second), and the responses of interest would continue to be victim traffic load and server CPU utilization. A full statistical analysis (e.g. using Latin Hypercube Sampling) on an emulation-based experiment would be computationally intensive, so a study of various surrogate functions (e.g. Polynomial Chaos Expansion) and the amount of training data used to fit those functions would provide interesting insight into their suitability for efficiently generating statistics.

The full factorial design described in the previous section is useful because it provides data suitable for main effects and interaction effects studies. However, the experiment required 12 hours of wallclock time (on a single server) to iterate through all combinations. As the dimensionality of the parameter space and number of levels per parameter increases, fractional factorial designs will be needed to efficiently run the experiment. As described earlier, there is a rich body of research in fractional factorial experimentation for other domains, but it is unclear whether these results are valid for cyber experimentation. For example, cyber systems have discontinuous responses (e.g. due to IP packet fragmentation) and/or long-tailed distributions and rare events (e.g. race conditions) that would may require new strategies for sampling parameter values in a fractional factorial design.

Similarly, future work in optimization (especially "black box" optimization, where an external tool drives an experiment in an attempt to find local or global minima/maxima) would also need to account for discontinuities in responses. Traditionally, these tools have been coupled to simulation codes that model continuous behavior (e.g. fluid dynamics codes). However, discontinuous responses in cyber systems may cause the black box optimization tool to converge on the wrong result (e.g. if it is using gradient descent).

5. Summary

We have presented a broad terminology of concepts related to design of experiments, experimental design, sensitivity analysis, and uncertainty quantification. We also provided a demonstration of a full factorial design on a DNS amplification attack scenario. The analyses we presented (variable ranking by importance, analysis of variance main effects studies, and Pareto optimization) demonstrate a few of the things that can be learned from ensemble generation of cyber experiments.

We believe there are many more sophisticated analyses that can be performed. One example is a "multifidelity framework", where thousands of low fidelity runs (e.g. from a very coarse Emulytics model or from a network simulator) are used to augment a small number of high fidelity model runs to obtain lower-variance estimates on high-fidelity results than would be possible otherwise [4]. Another example is robust design or design under uncertainty, where the ensemble generation is done in a nested framework: an outer optimization loop chooses the design settings and an inner UQ loop chooses the stochastic parameters to vary to obtain an expected value (for example) which can be optimized. There is a large and growing interest in various types of surrogate models (including machine learning) that may have a role in full-scale cyber experiments which are very expensive to run. Finally, we expect that sampling methods and dimension reduction methods may be customized to address the large number of input variables which are present in full-scale cyber experiments.

5. Acknowledgment

This work is supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

6. References

[1] AIAA Standards: *Guide for the Verification and Validation of Computational Fluid Dynamics Simulations* (AIAA G-077-1998(2002)) Computational Fluid Dynamics Committee. https://doi.org/10.2514/4.472855

[2] ASME V&V 20-2009 *Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer*. https://www.asme.org/products/codes-standards/v-v-20-2009-standard-verification-validation

[3] Diegert, K., Klenke, S., Novotny, G., Paulsen, R., Pilch, M. and T. Trucano. *Toward a More Rigorous Application of Margins and Uncertainties within the Nuclear Weapons Life Cycle – A Sandia Perspective*. Sandia Technical Report SAND2007-6219.

[4] Geraci, G., L.P. Swiler, J. Crussell, and B. Debusschere. *Exploration of Multifidelity Approaches to Uncertainty Quantification in Network Applications*. UNCECOMP ECCOMAS 2019. Thematic Conference on Uncertainty Quantification in Computational Sciences and Engineering. SAND2019-3274C.

[5] Ghanem, R. And P. Spanos, *Stochastic Finite Elements: A Spectral Approach*. New York, New York: Springer Verlag (2002).

[6] Hedayat, A.S., Sloane, N.J.A., and J. Stufken. *Orthogonal Arrays: Theory and Applications*. Springer Series in Statistics, 1999.

[7] Helton, J.C. *Conceptual and Computational Basis for the Quantification of Margins and Uncertainty*. Sandia Technical Report 2009-3005.

[8] Jones, S.T., Gabert, K. G., and T. D. Tarman. *Evaluating Emulation-based Models of Distributed Computing Systems*.SAND2017-10634. https://www.osti.gov/biblio/ 1398865-evaluating-emulation-based-models-distributed-computing-systems.

[9] Maricq, A., Duplyakin, D., Jiminez, I., Maltzahn, C., Stutsman, R. and R. Ricci. *Taming Performance Variability*. 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI). 2018

[10] Mirkovic, J., Bartlett, G. and J. Blythe. *DEW: Distributed Experiment Workflows*. USC Information Sciences. Proceedings from USENIX/CSET 2018 Conference.

[11] Morgan, M. G. and M. Henrion. Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis. Cambridge University Press, 1990.

[12] National Research Council. Assessing the Reliability of Complex Models: Mathematical and Statistical Foundations of Verification, Validation, and Uncertainty Quantification. Washington, DC: The National Academies Press, 2012. https://doi.org/10.17226/13395.

[13] Oberkampf, W.L. and C.J. Roy. Verification and Validation in Scientific Computing. Cambridge University Press, 2010.

[14] Rasmussen, C.E. And C.K.I. Williams, Gaussian Processes for Machine Learning. MIT Press (2006).

[15] Sacks, J., W.J. Welch, T.J. Mitchell, and H.P. Wynn. Design and analysis of computer experiments. Statistical Science, 4(4):409–435, 1989.

[16] Saltelli, A., Chan, K., Scott, E.M. Sensitivity Analysis. New York: Wiley; 2000.

[17] Saltelli, A., Tarantola, S., Campolongo, F., Ratto, M. Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models. New York: Wiley; 2004.

[18] Santner, T., B. Williams, And W. Notz, The Design and Analysis of Computer Experiments. New York, New York: Springer (2003).

[19] Seber, G.A.F. And C.J. Wild, Nonlinear Regression. New York, New York: Wiley & Sons (2003).

[20] Simpson, T.W., V. Toropov, V. Balabanov, and V. F.A.C. Design and analysis of computer experiments in multidisciplinary design optimization: A review of how far we have come or not. in Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference. 2008. Victoria, British Columbia, Canada. AIAA Paper 2008-5802.

[21] Xiu, D., Numerical Methods for Stochastic Computations: A Spectral Method Approach. Princeton University Press (2010).