



**Calhoun: The NPS Institutional Archive**  
**DSpace Repository**

---

Faculty and Researchers

Faculty and Researchers' Publications

---

2021

# Data Farming: The Meanings and Methods Behind the Metaphor

Sanchez, Susan M.

Operational Research Society

---

Sanchez, Susan. "Data Farming: The meanings and methods behind the metaphor." Conference Paper, Operational Research Society Simulation Workshop (2021).  
<http://hdl.handle.net/10945/69157>

---

This publication is a work of the U.S. Government as defined in Title 17, United States Code, Section 101. Copyright protection is not available for this work in the United States.

*Downloaded from NPS Archive: Calhoun*



<http://www.nps.edu/library>

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

**Dudley Knox Library / Naval Postgraduate School**  
**411 Dyer Road / 1 University Circle**  
**Monterey, California USA 93943**

## DATA FARMING: THE MEANINGS AND METHODS BEHIND THE METAPHOR

*Susan M. Sanchez*

Naval Postgraduate School  
Monterey, California, USA  
ssanchez@nps.edu

### ABSTRACT

Data farming captures the notion of purposeful data generation from simulation models. The ready availability of computing power has fundamentally changed the way simulation and other computational models can be used to provide insights to decision makers. Large-scale designed experiments let us grow the simulation output efficiently and effectively. We can explore massive input spaces, use statistical and visualization techniques to uncover interesting features of complex response surfaces, and explicitly identify cause-and-effect relationships. Nonetheless, there are many opportunities for research methods that could further enhance this process. I will begin with a brief overview of key differences between physical and simulation experiments, as well as current data farming capabilities and their relationship to emerging techniques in data science and analytics. I will then share some thoughts about opportunities and challenges for further improving the state of the art, and transforming the state of the practice, in this domain.

**Keywords:** Simulation, Experimental Design, Visualization

### 1 INTRODUCTION

We live in a world bombarded by data. The term ‘data mining’ is ubiquitous in the literature, while ‘data analytics’ and ‘data science’ have skyrocketed in popularity in recent years. Much of this digital dust is collected automatically—by our communication technology, sensors in the environment, cookies placed on websites, wireless devices comprising the internet of things, and more. Some of these observational data sources can be used to characterize input distributions for stochastic simulation models, either by fitting distributions from which pseudo-random numbers are generated, by bootstrapping samples from the empirical distributions, or used in data-driven simulation models to affect real-time system intervention and control. Yet this type of data is observational by nature, and so has limitations. Simulation output data, by contrast, is available only after the simulation is run, where a ‘data farming’ metaphor is more appropriate. Consider this description: “Real-world farmers cultivate the land to maximize their yield. They manipulate the environment to their advantage by using irrigation, pest control, crop rotation, fertilizer, and more. Small-scale designed experiments can help them to determine whether these treatments are effective. Similarly, data farmers manipulate simulation models to their advantage—but using large-scale designed experimentation. This allows them to learn more about the simulation model's behavior in a structured way. In this fashion, they ‘grow’ data from their models, but in a manner that facilitates identifying the useful information. For large-scale simulation experiments, this often results in data sets that, while big, are far smaller than what would be needed to gain insights if the results were observational (i.e., obtained using *ad hoc* or randomly generated combinations of factor settings). Data generated prospectively from designs is also better, in the sense that it lets us identify root cause-and-effect relationships between the simulation model input factors and the simulation output.” (Sanchez 2018).

Simulation is not the only community to use the data farming metaphor. As a noun, data farm may refer to a large bank of connected computers used to process and store data, host web services,

provide access for scientific computing, and more. As a verb, data farming has been used as a metaphor for dealing with big data in non-simulation contexts: see, e.g., Kusiak (2006) for enhancing industrial data for decision-support purposes, or Mayo et al. (2016) for improving patient outcomes in healthcare settings. These data farming approaches attempt to improve the collection, storage, maintenance, and retrieval of observational data so it is faster and easier to harvest insights. While some effort has been made to address causality from observational datasets (Pearl 2009), we can distinguish the simulation data farming view as one of generating and analyzing inferential big data, in contrast to methods for curating and analyzing observational big data.

Schruben (2017) asserts that “model is a verb” for simulation professionals. Likewise, data farming is a verb from the simulation perspective we use in this paper.

## 1.1 Background Terminology and Notation

*Factors* are inputs (or functions of inputs) to a simulation model that are purposefully varied at different levels when growing the data from a simulation experiment. An *experiment design* for  $k$  factors is an  $n$  by  $k$  matrix or table where each column specifies the levels or settings for a single factor and each row specifies the combination of factor settings to be used. We refer to the rows as *design points*, they might also be called runs or trials in other literature.

*Features* are characteristics of the *response surface* that maps the inputs to the simulation outputs. A statistical or analytical model of our simulation model’s I/O behavior is called a *metamodel* because it is a model of a model. Many types of metamodels are possible, including partition trees (also known as classification and regression trees), multiple regression metamodels, logistic regression metamodels, Gaussian process metamodels, and more.

*Flexibility* is an important consideration when embarking on a data farming study because the types of designs used to grow the data will affect the types of metamodels we can fit, and the types of questions we can answer. In the data farming context, we are proponents of ‘thinking big’ in terms of the number and types of factors, the number of outputs and breadth of their response surface behaviors, and the types of analysis tools and methods that can be applied to the output data.

## 2 MEANINGS AND METHODS

In the rest of this paper, we will focus on data farming (the verb) as a metaphor for simulation studies. We will describe the meanings of several subcomponents of this metaphor, and present some practical data farming methods, with the goal of encouraging the readers to incorporate data farming into their future simulation studies.

### 2.1 Cross Fertilization

In our experience, data farming is most effective when it is a collaborative effort (NATO 2014). Stakeholders in the problem domain help ground the data farming effort and ensure that it does, in fact, address questions and provide insights that are useful and interesting to decision makers. Simulation modelers bring a variety of expertise. At early stages of a simulation study, their conceptual modelling skills may help scope the project so the simulation model is neither overly simplified nor overly complex for its intended purposes. They catch logical misconceptions that might invalidate the results or interpretation, such as a user who does not realize that different random number seeds lead to different results, or that modelling a queue as capacitated vs. uncapacitated will yield different results. Problem domain experts are key players in the model validation process.

One practical piece of advice is to ask all stakeholders to jot down a few key expectations, such as “What do believe the three most important factors will be? How will they affect the response?” Done early, this may lead to discussions that help frame the conceptual model and make sure there is a common understanding of its component, especially if the stakeholders have different backgrounds and expertise. Done before running the experiment, this helps ensure that the factors, their ranges or settings, and the experimental design used will be suitable for addressing the initial questions—although it is better to think of experimentation and analysis as an iterative process instead of a single event. Done after the data have been generated but before conducting analysis, this may help clarify

whether or not the results are surprising. Ultimately, when a surprising result is found, it should either lead to a bug being fixed (model verification) or intuition being changed (model validation).

## 2.2 Sowing the Seeds

Design of experiments (DOE) can be viewed as sowing the seeds for successful data farming, and brings tremendous capabilities to simulation studies. There are several reasons for this. First, experimentation is a straightforward way of establishing cause-and-effect. By purposefully varying factors using a good design, we can observe what (if any) effects they have on the responses—at least within the context of our simulation model. Varying multiple factors simultaneously is the only way to reveal interactions effects, varying factors at many levels in a space-filling design provides analysis flexibility, and using a good experimental design is absolutely required. What constitutes a bad design? A one-factor-at-a-time design is bad because it does not reveal any interactions. A design with high correlations among factors is bad design because it means that factor effects are confounded, so there is no unique way to determine which factors impact the response. A design that cannot be executed in the time required is a bad design because it means the decision maker cannot leverage insights from the study. A design that ignores factors simply to reduce the number of design points is bad because it drastically limits the potential insights that could be gained.

There are many good experimental designs, but some are more suitable for physical experiments or deterministic computer experiments than for stochastic simulation experiments. Here are a few that we recommend, use often, and are readily available for you to use in your next data farming experiment:

- Nearly orthogonal Latin hypercubes (NOLHs),
- Nearly orthogonal-and-balanced (NOAB) designs,
- Resolution V fractional factorials (R5FFs),
- Resolution V central composite designs (R5CCDs), and
- Frequency based designs (R5FBDs).

More details of these designs and their characteristics and applicability appear in the Appendix.

As a practical tip, follow the links in the Appendix to download the software and run a data farming experiment. The tutorial paper by Sanchez, Sanchez, and Wan (2020) discusses both design and analysis considerations in more depth.

## 2.3 Pest Control

In Section 2.1 we described how stakeholders' predictions of which factors will be most important can be helpful in verification and validation (VandV) efforts. A large-scale sensitivity analysis is a much broader and more rigorous way of stress-testing a simulation model.

This debugging effort also reinforces the view that model is a verb. We should not separate the process of modelling and experimentation, they enhance each other. It is better to continually experiment as you go along and build a model, catching at least some of the bugs earlier, than waiting until the end. Experimentation can also help the modeller avoid adding unnecessary model detail if it becomes clear that variation in certain model subcomponents is dampened by the system, so additional complexity is not warranted. For example, if varying a deterministic setup time for a station in a job shop between 15 minutes and 30 minutes does not yield a noticeable difference in overall throughput, then it would not be worthwhile to expend effort to create a stochastic setup time that varies over that same range.

At any stage, a practical way of proceeding is to begin with a baseline design point. Set the ranges for each quantitative factor a small percentage (say, 5% or 10%) above and below the baseline (if the baseline is at the lowest or highest level of interest, expand the range in only one direction). Run designed experiments regularly during the model-building process. This also means the model you're making will be data farmable, which will save you the time of having to restructure the finished model or create a data farming wrapper to facilitate experimentation. It also means you will easily identify situations where the model behaves strangely or stops working. We have often found it possible to diagnose and track down errors by using such a method. For example, in one experiment we varied thirty simulation inputs that we had previously left unchanged, and found that the

simulation failed to run part of the time. A few splits of a partition tree isolated the problem to an interaction between two factors that could lead to a buffer overflow. The bug could then be corrected.

## 2.4 Harvesting Efficiently

Automation is a key enabler of data farming, since there are many repetitive tasks. A little work up front makes life much easier down the road.

If you are just getting started on data farming, you may find it helpful to use some of the run control scripts in the `datafarming` Ruby gem described in Section 2.2. These are scripts that allow you to run any simulations that can be run from the command line (such as simulations written in python, Matlab, R, java, or similar languages) and .

When you are ready, parallel computing can easily be leveraged for purposes of data farming. Each run (a single replication of a single design point) is a self-contained simulation that can be sent off to a core, with the data consolidated once all runs are complete. Software such as HTCondor at <https://research.cs.wisc.edu/htcondor/> [accessed 1 March 2021] can be used to farm jobs out to multiple cores, either on a single multicore machine or on a computing cluster. SESSL at <http://sessl.org> [accessed 1 March 2021] is another software language set up to facilitate experiments for a variety of simulation modelling platforms (Ewald and Uhrmacher 2014; Warnke and Uhrmacher 2018). For more about the nuts and bolts of data farming, see Sanchez and Sanchez (2017). If your models are set up to be data farmable from the start, running the data farming experiments is straightforward—and you will never want to go back to manual experimentation.

## 2.5 Maximizing Yield

By maximizing yield, we mean gaining as much knowledge and insight as we can from our simulation study to inform decision makers. This may be insight about the simulation model’s behavior itself, or about a real-world situation that we are simulating. If the model’s intended use is to assist decision makers on important and complex questions, then we should ‘think big’ in terms of the insights that might be gained. Decision makers attempting to address complex problems are not likely to be interested in answers to simple questions. Given the time and effort that can be spent to conceptualize and implement a simulation model, make sure that effort is put to good use. Data farming is a way to make your simulation model work for you!

Think of robustness as you plan your data farming experiments (Sanchez and Sanchez 2020). Robustness is a structured way to guard against making unwarranted assumptions. Factors in your data farming experiment can be differentiated as decision factors, noise factors, and artificial factors. Decision factors are those that can be controlled in the real-world setting for which the simulation is based. Noise factors are those that either cannot be controlled, or can be controlled only at great cost or difficulty, in the real world. Artificial factors are specific to the simulation environment, such as the warm-up period for steady-state simulations; choices of random number generators, seeds, or streams; run lengths; time intervals for discrete-time simulations; and more. Including artificial factors in a data farming experiment may yield insights about using simulation for real-time control. Including both decision and noise factors makes it more likely that recommended solutions will work well for a broad range of situations that might arise in practice, even if these are not optimal solutions for any particular setting. A robustness perspective can also be used to ascertain whether certain model assumptions, such as input distribution shapes, lead to substantively different recommendations. The current combination of computational power and modelling platforms and paradigms helps simulation modelers to reduce so-called ‘Type III errors’ of solving the wrong problem (Mitroff and Featheringham 1974). Seeking robust solutions aids this process.

## 2.6 Reaping the Benefits

Once we have generated an inferential big data set from our data farming experiment, what do we do with it? We have found that just as “having” big data from the internet meant that companies found new and exciting things to do with it, having big data from simulation experiments offers the opportunity for new and interesting ways of looking at the results (Elmegreen, Sanchez, and Szalay 2014). These include a wide variety of metamodeling and visualization techniques.

Theoretically, every one of the inputs should affect at least one response in some way. If not, there is something wrong with either the conceptual model (e.g., we have added unnecessary detail or left out connections) or its implementation (the code contains bugs). However, even if all factors have some ‘true’ effect, that does not mean they are all equally important. Data farming can help us identify the factors or interactions that are key drivers of performance over the region of factor exploration. Consequently, when constructing metamodels we may end up excluding factors or terms that are statistically significant—either because they are dwarfed by other factors or terms that have much stronger effects, or because their effects, while statistically significant, are not of practical interest given the region of interest for this particular experiment.

Some features are best revealed by graph-analytic techniques: see, e.g., Feldkamp Bergmann and Strassburger (2015, 2020), Matković, Gracanin, and Hauser (2018), or Sanchez (2020) for examples drawn from simulation experiments. Past data farming studies have helped save lives, time, money, and the environment; improve algorithms; and facilitate thoughtful discussions around modelling human behaviors and interactions.

## **2.7 Serving the Community**

Our metaphor involves farming, not gardening. In the real world, both might be used to grow vegetables (or herbs, or flowers)—but a garden is a small plot intended for private use, while a farm is a larger enterprise that grows crops for others. This sense of distributing results to a large community of stakeholders, rather than simply generating the insights for ourselves, is important. The sense of scale also matters. We have over a trillion times the computing power at our fingertips than was used to first put a man on the moon (Lucas et al. 2015). How are we leveraging this power? Are our methods of building and analyzing simulation models keeping pace?

## **3 CONCLUDING THOUGHTS**

Going forward, there are many opportunities for advancing the theory, the practice, and the applications of simulation. This work can be worthwhile, rewarding, fascinating, and fun! We hope the data farming metaphor helps researchers think broadly about how their talents and interests might grow the capability in one or more of these areas, and anticipate the needs that practitioners will face in the future. We hope this metaphor resonates with practitioners, allowing them to reap immediate benefits by using a data farming approach for their next simulation study. We hope that the breadth and depth of insights that can be gleaned will help decision makers in the public and private sectors turn to simulation as a means of obtaining useful, robust, and actionable recommendations to address the complex problems they face.

Our global simulation community has opportunities to make differences in all these dimensions. The COVID pandemic of the past year is but one striking example of how useful and important it can be to gain insights from modelling and simulation. Virtual experiments have helped facilitate timely decision making for numerous types of systems at a variety of levels, from procedures for administering tests and vaccines, to creating new layouts and patient flows for specific healthcare facilities, to policy recommendations at local, regional, or national levels intended to contain and halt the spread of the disease. The pandemic response also makes it clear that modelling and simulation are not enough. Our simulation community must continue to strengthen its ties and outreach to other communities—sharing with them, listening to them, and learning from them—to reach our full potential and help address the major challenges our world now faces.

## **ACKNOWLEDGMENTS**

Thanks to the many people I have had the pleasure of discussing with and working on these concepts over the years. Department of Defense (DoD) Distribution Statement: Approved for public release; distribution is unlimited. The views expressed in this document are those of the author and do not necessarily reflect the official policy or position of the Navy, DoD, or the U.S. Government.

## APPENDIX

There are many potential classes of designs available. Here are a few that we recommend, use regularly, and are readily available for you to use yourself. Most are components of the `datafarming 1.4.0` Ruby gem that can be found at <https://rubygems.org/gems/datafarming> [accessed 1 March 2021]; the README file has instructions for installing and running the gem on Windows, MacOS, or Linux systems. See [https://bitbucket.org/paul\\_j\\_sanchez/datafarmingrubyscripts/src/master/](https://bitbucket.org/paul_j_sanchez/datafarmingrubyscripts/src/master/) [accessed 1 March 2021] to view or download the source code.

- ***Nearly orthogonal Latin hypercubes (NOLH)***. These are space-filling designs suitable for quantitative factors that are continuous-valued or discrete-valued with many levels. The maximum absolute pairwise correlation between any two columns is less than 0.05. Several base design sizes (maximum factors  $k$ , number of design points  $n$ ) can be used: those currently coded are (7,17), (11,33), (16,65), and (22,129) (Cioppa and Lucas 2007); (29,257) (Hernandez, Lucas, and Carlyle 2012), and (100,512) (Vieira et al. 2013). A shift-and-stack approach can generate larger designs with improved space-filling behavior for any  $k$  between 2 and 100.
- ***Nearly orthogonal-and-balanced (NOAB) designs***. These are suitable ‘as is’ for quantitative factors and discrete-valued factors with 2 to 11 levels, with maximum absolute pairwise correlation of 0.0347 between any two columns. Nearly-balanced means that the levels of any particular discrete-valued factor appear in roughly equal numbers of design points. A customizable 512-dp (design point) NOAB allows the analyst to create a design involving up to 20  $m$ -level factors ( $m = 2, 3, \dots, 11$ ) and 100 continuous-valued factors. With a little extra care, the discrete-valued columns can be used for qualitative factors as well. Also, the entire design can be shifted-and-stacked if the shift-and-stack is applied separately to each  $m+1$  groups of columns: one group for the  $m$ -level factors) ( $m = 2, 3, \dots, 11$ ) and one group for up to 50 continuous-valued quantitative factors.
- ***Resolution V fractional factorials (R5FF)***. These orthogonal designs are suitable for any mix of two-level factors, either qualitative or quantitative. They are not space-filling, but they have the property that all main effects, all quadratics, and all two-way interactions can simultaneously be estimated. Design sizes are powers of two. The design generators can be stored efficiently, and result in design sizes that are powers of two. Some examples are  $2^2=4$  dps for  $k=2$ ,  $2^{20-11}=512$  for  $k=20$ ,  $2^{50-38}=4096$  for  $k=50$ , and  $2^{120-105}=32768$  for  $k=100$ . Applying shift-and-stack to these designs does not improve space-filling, but it does increase the number of corner points sampled.
- ***Resolution V central composite designs (R5CCD)***. These orthogonal designs are suitable for quantitative factors, and they have the property that all main effects and all two-way interactions can simultaneously be estimated. They augment the R5FFs with one center and  $2k$  star points. This results in three levels per factor if the star points are placed on the faces of the hypercube, or five levels per factor if all non-center points are an equal distance from the center (a rotatable CCD). The improved space-filling behavior provides greater metamodel flexibility. Metamodels with quadratic terms can be fit from the output data for both types of CCDs. Metamodels could contain cubic or quartic terms for the rotatable CCDs.
- ***Frequency based designs (R5FBD)***. These orthogonal designs are suitable for quantitative factors, and have the property that all main effects, all quadratics, and all two-way interactions can simultaneously be estimated. Factor levels can be viewed as oscillating sinusoidally at carefully selected frequencies as a function of the design point. R5FBDs have a smaller proportion of dps in the interior of the sampling region than NOLHs, but a larger proportion than R5FFs. Some examples of design sizes are 13 for  $k=2$ , 1673 for  $k=20$ , 17761 for  $k=50$ , and 115434 for  $k=100$ . Applying a shift-and-stack approach to these designs improves their space-filling behavior.

Another straightforward way of creating a design that contains both qualitative and quantitative factors is to create two separate designs  $D_1$  (for  $k_1$  factors in  $n_1$  dps) and  $D_2$  (for  $k_2$  factors in  $n_2$  dps) and then *crossing* them, obtaining a design for  $k_1+k_2$  factors in  $n_1n_2$  dps. A crossed design is typically much larger than a single combined design (such as a NOAB) so a combined design is usually preferred if either  $k_1$  or  $k_2$  is large.

Other websites for obtaining data farming software and designs include

- The `datafarming` Ruby gem has self-documenting scripts for design generation, design scaling, and data farming run control. The README file has instructions for installing and running the gem on Windows, MacOS, or Linux systems. [<https://rubygems.org/gems/datafarming> accessed 31 January 2021].
- Source code for the `datafarming` Ruby gem can be viewed or downloaded from [[https://bitbucket.org/paul\\_j\\_sanchez/datafarmingrubyscripts/src/master/](https://bitbucket.org/paul_j_sanchez/datafarmingrubyscripts/src/master/) accessed 31 January 2021].
- The Naval Postgraduate School's *SEED Center for Data Farming* website at has downloadable spreadsheets (such as the customizable 512-dp NOAB) and links to other software [<https://harvest.nps.edu> accessed 31 January 2021].
- The R package `FrF2Large` also has code for generating the R5FF designs [<https://rdrr.io/cran/FrF2/man/FrF2Large.html> accessed 31 January 2021].

There are several other R packages that create designs. Many commercial statistical software packages, and some simulation modelling platforms, also have design-generating capabilities.

## REFERENCES

- Cioppa T M and Lucas T W (2007). Efficient nearly orthogonal and space-filling Latin hypercubes. *Technometrics* **49(1)**: 45-55.
- Elmegreen B E, Sanchez S M, and Szalay A (2014). The future of computerized decision making. In: Tolk A, Diallo S D, Ryzhov I O, Yilmaz L, Buckley S, and Miller J A (eds). *Proceedings of the 2014 Winter Simulation Conference*. IEEE, Piscataway, New Jersey, pp 943-949.
- Ewald R and Uhrmacher A (2014). SESSL: A domain-specific language for simulation experiments. *ACM Transactions on Modeling and Computer Simulation* **24(2)**: 11:1-11:25.
- Feldkamp N, Bergmann S, and Strassburger S (2015) Visual analytics of manufacturing simulation data. In: Yilmaz L, Chan W K V, Moon I, Roeder T M K, Macal C, Rossetti M D (eds) *Proceedings of the 2015 Winter Simulation Conference*. IEEE, Piscataway, New Jersey, pp 779–790.
- Feldkamp N, Bergmann S, and Strassburger S (2020). Knowledge discovery in simulation data. *ACM Transactions on Modeling and Computer Simulation* **30(4)**: 24:1-24:25.
- Hernandez A S, Lucas T W, and Carlyle M (2012). Constructing nearly orthogonal Latin hypercubes for any nonsaturated run-variable combination. *ACM Transactions on Modeling and Computer Simulation* **22(4)**: 20:1-20:17.
- Lucas, T W, Kelton W D, Sanchez P J, Sanchez S M, and Anderson B L (2015). Changing the paradigm: Simulation, now a method of first resort. *Naval Research Logistics* **62**: 293-305.
- Pearl J (2009). *Causality: Models, Reasoning, and Inference (2nd ed.)*. Cambridge University Press, New York, NY.
- Kusiak A (2006). Data farming: Concepts and methods. In: Triantaphyllou E and Felici G (eds). *Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques*. Springer US: Boston, Massachusetts, pp 279–304.
- Matković K, Gracanin D, and Hauser H (2018). Visual analytics for simulation ensembles. In: Rabe M, Juan A A, Mustafee N, Skoogh A, Jain S, and Johansson B (eds). *Proceedings of the 2018 Winter Simulation Conference*. IEEE: Piscataway, New Jersey, 321-335.
- Mayo C S, Kessler M L, Eisbruch A, Weyburne G, Feng M, Hayman, J A, ... Ten Haken R K (2016). The big data effort in radiation oncology: Data mining or data farming? *Advances in Radiation Oncology* **1**: 260–271.



- Sanchez S M (2018) Data farming: Better data, not just big data. In: Rabe M, Juan A A, Mustafee N, Skoogh A, Jain S, and Johansson B (eds). *Proceedings of the 2018 Winter Simulation Conference*. IEEE: Piscataway, New Jersey, pp 425–439.
- Sanchez P J and Sanchez S M (2019). Orthogonal second-order space-filling designs with insights from simulation experiments to support test planning. *Quality and Reliability Engineering International* **35(3)**: 854-867
- Sanchez S M (2020). Data farming: Methods for the present, opportunities for the future. *ACM Transactions on Modeling and Computer Simulation* **30(4)**: pp 22:1-22:30.
- Sanchez S M and Sanchez P J (2017). Better big data via data farming experiments. In: Tolk A, Fowler J., Shao G, and Yücesan E (eds). *Advances in Modeling and Simulation: Seminal Research from 50 Years of Winter Simulation Conferences*. Springer International Publishing AG: Cham, Switzerland, pp 17-26.
- Sanchez S M and Sanchez P J (2020). Robustness revisited: Simulation optimization viewed through a different lens. In: Bae K-H, Feng B, Kim S, Lazarova-Molnar S, Zheng Z, Roeder T, and Thiesing R (eds). *Proceedings of the 2020 Winter Simulation Conference*. IEEE: Piscataway, New Jersey, pp 60-74.
- Schruben, L W (2017). Model is a verb. In: Tolk A, Fowler J., Shao G, and Yücesan E (eds), *Advances in Modeling and Simulation: Seminal Research from 50 Years of Winter Simulation Conferences*. Springer International Publishing AG: Cham, Switzerland, pp 17-26.
- Vieira H, Sanchez S M, Kienitz K H K and Belderrain M C M (2013). Efficient, nearly orthogonal-and-balanced, mixed designs: An effective way to conduct trade-off analyses via simulation. *Journal of Simulation* **7(4)**: 264-275.
- Warnke T and Uhrmacher A M (2018). Complex simulation experiments made easy. In: Rabe M, Juan A A, Mustafee N, Skoogh A, Jain S, and Johansson B (eds). *Proceedings of the 2018 Winter Simulation Conference*. IEEE: Piscataway, New Jersey, pp 410-424.

## AUTHOR BIOGRAPHY

**SUSAN M. SANCHEZ** is a Distinguished Professor in the Operations Research Department at the Naval Postgraduate School, with a joint appointment in the Graduate School of Defense Management. Her research interests include experimental design, data-intensive statistics, and robust selection, with applications to simulation experiments, manufacturing, health care, and military operations. She is Co-director of NPS's *SEED Center for Data Farming*, which was established to advance the use of simulation experiments and efficient designs to provide decision makers with timely insights. <https://harvest.nps.edu>