Proceedings of the 2017 Winter Simulation Conference
W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds.

MODEL ALIGNMENT USING OPTIMIZATION AND DESIGN OF EXPERIMENTS

Alejandro Teran-Somohano
Alice E. Smith

Department of Industrial and Systems Engineering

Auburn University
3301 Shelby Center
Auburn, AL 36849 USA

Levent Yilmaz

Department of Computer Science and Software Engineering
Auburn University
3101 Shelby Center
Auburn, AL 36849 USA

ABSTRACT

The use of simulation modeling for scientific tasks demands that these models be replicated and independently verified by other members of the scientific community. However, determining whether two independently developed simulation models are "equal," is not trivial. Model alignment is the term for this type of comparison. In this paper, we present an extension of the model alignment methodology for comparing the outcome of two simulation models that searches the response surface of both models for significant differences. Our approach incorporates elements of both optimization and design of experiments for achieving this goal. We discuss the general framework of our methodology, its feasibility of implementation, as well as some of the obstacles we foresee in its generalized application.

1 INTRODUCTION

The present work is part of a larger research endeavor focused on simulation model transformation across different platforms. The overarching goal is to support the replicability and reproducibility of simulation experiments for scientific discovery. Model transformation is not, of itself, sufficient to achieve this goal. It is also necessary to verify that the transformed model properly mimics the behavior of the original. But what does it mean for a model to mimic another model? In this paper, we consider two models to be equal if the response surfaces of both models are equal. Due to the stochasticity of many simulation models, the actual response surfaces will most likely differ in absolute terms, so statistical methods are required to determine whether the differences are statistically significant. In other words, the two response surfaces are considered equal if, for each input parameter combination, the statistical distribution of the responses is equal.

In consequence, determining whether two models are the same requires comparing their response surfaces. For any real application, a brute force approach—one where each parameter combination is evaluated and compared—is not only undesirable, but practically impossible due to the size of the parameter space, and the computational effort required to explore the response surfaces in their entirety. That is, evaluating the models at each and every point yields little information about the response surface per model evaluation, so to obtain the requisite confidence many replications are required. Because this is computationally impossible, we need a method that gains additional information about the response surface with each model run.

What we are looking for, then, is an *intelligent* search method for exploring the response surfaces. For instance, one that is aimed at finding potential differences (if they exist) between the two surfaces and focusing the comparison to those points.

To find these points of potential difference, we propose using an optimization algorithm. Optimization algorithms are essentially search algorithms that seek a combination of parameters that

optimize a certain objective function. In this case, we are searching for those points where the difference between both response surfaces is maximized. That is, instead of comparing the two responses over the entire parameter space, we identify and concentrate on those areas where it is most likely that the responses differ.

Once these points are found (there can be more than one), we can run a designed experiment around them to sample from both response surfaces and gather enough information to perform a statistical analysis that can help us determine whether the responses from both models are drawn from the same distribution or not. Note that we confine ourselves to model alignment of two simulation models in this paper but the approach can be readily extended to three or more model comparisons.

This paper discusses such an approach in greater detail, and outlines some of the obstacles and potential applications of such approach. The paper is organized as follows: Section 2 goes over the concept of model alignment, how it has been used for model comparison, and some of its limitations. We also explain how our approach represents an extension of the model alignment methodology. Section 3 presents our methodology, including a discussion of some of the important decisions that have to be made to implement it. Section 4 describes a proof-of-concept demo that we developed to test the ideas presented in this paper. Finally, Section 5 covers the limitations of our approach, our conclusions, and some future work.

2 BACKGROUND

The methodology described in this paper combines aspects of model alignment and design of experiments. In this section, we discuss these in greater detail, with a particular emphasis on how our approach extends those found in the literature.

2.1 Model Alignment

Model alignment is defined as the process "needed to determine whether two models can produce the same results" (Axtell et al. 1996). It is an essential process for model replicability and reproducibility. It is also referred to as "model docking." In our particular case, the investigation is motivated by some of our previous research into model transformation. Consider the situation where a simulation model was converted from one platform to another. How can we know whether the transformation produced a model that is enough like the original in its modeling ability? We need some method for comparing the outcomes of both models and determining whether they are equal. If the models are deterministic, this is fairly straightforward, either they yield the same result, or they do not. If, however, they are stochastic, then we must rely on statistical techniques that can tell us whether both models produce the same distribution of results. We are concerned, then, with verifying the success or failure of a model replication. Wilensky and Rand (2007) identify six dimensions along which a model and its replicates can differ: (1) time of the implementation, (2) hardware used, (3) language of implementation, (4) toolkits, (5) algorithms, and (6) authors. In the context of our research, the models differ along several dimensions, but particularly dimension (3). We want to verify whether a model that has been converted from one language (or simulation platform) to another is a sufficient replicate of the original. A successful replication of a model (a successful alignment), means that the output of the original model and its replicate are "similar enough." This requires a criterion for judging whether they are similar enough, which we, following Wilensky and Rand (2007) will term the *replication standard*. Of particular interest is the discussion provided in Axtell et al. (1996) about this standard of model equivalence. They identify three possible criteria: (1) numerical identity, (2) distributional equivalence, and (3) relational equivalence. Numerical identity cannot be expected in simulation models with stochastic elements in them. By distributional equivalence is to be understood that both models yield distributions of results that are not statistically distinguishable, that is, the outcome of both models is "drawn from the same distribution." Lastly, relational equivalence implies that both models produce the same internal relationship among their

results, for instance, if the response increases by increasing an input in one model, it does the same in the other (Axtell et al. 1996).

Model alignment was first introduced by Axtell et al. (1996). In their research, they compared two models, one of which was thought to be a more general version of the other. Their goal was to compare a simplified version of the more general model, and evaluate whether it produced the same outcome. To do this, they ran the model at the same parameter combinations as the experiment that had been performed on the original model when it was first published. This model comparison is used as a case study and as a backdrop for a larger discussion on what it means for two simulation models to "be the same," as well as to discuss other issues faced when aligning two models.

Wilensky and Rand (2007) describe their experience trying to replicate an agent-based model, and offer a series of recommendations for facilitating model replicability. Edmonds and Hales (2003) also describe their experience replicating an agent-based model and using alignment methods to evaluate their success. In particular, they describe how they detected some errors in the implementation of the original model, so the use of model alignment is not only a tool for replicability, but also for verification and validation. Miodownik et al. (2010) also aligned two agent-based models and used the alignment process to detect not only programming errors, but also hidden assumptions. Model alignment has also been used to evaluate whether model parallelization has been done successfully (Fachada et al. 2016), and to compare modeling toolkits (Xu, Gao, and Madey 2003). Another interesting perspective on the topic is provided by Will and Hegselmann (2008) who document a failed attempt at replicating a simulation model, and followed it up with a detailed analysis of the code and assumptions made in the original model, and which were not properly specified, leading to the failed alignment (Will 2009).

Fachada et al. (2017) discuss some of the limitations of the model alignment methods proposed by other authors, and propose using Principal Component Analysis (PCA) to align different implementations of the same model. They argue that traditional approaches rely on statistical summaries representative of each output (for instance, the average of an output), called focal measures (FM), for their comparison. These FMs are selected by the model designer and are always dependent on the model. Their approach, on the other hand, converts all of the model's outputs into a set of linearly uncorrelated measures that are then analyzed. This removes the dependence on the FMs, making it a model-independent method. In addition to that advantage, this approach also detects automatically which output features best explain differences between implementations, does not depend on the output's distributional properties, and works directly on simulation output.

What stands out in all these uses of model alignment is that the comparison between models is limited to those parameter combinations included in a previously defined experiment. As a result, this only allows them to affirm that the models are aligned along the parameter combinations used in the experiment. It is possible that they are not aligned along other parameter combinations, so even when Fachada et al. (2017)'s approach is model-independent, it is not independent of the input parameter combinations used to run the model.

2.2 Experimental Design

Since our goal is to evaluate whether two models are "the same," it is not enough to compare them over a pre-defined set of parameter input values. We need to verify whether they produce the same output over the entire input parameter space, or a surrogate for this. Space-filling experimental designs are meant to do this. The most well-known space-filling design is the Latin hypercube (McKay, Beckman, and Conover 1979), though there are many other designs available (Pronzato and Müller 2012). These designs "seek to find a model that approximates the true response surface over a much wider range of the design variables, sometimes extending over the entire region of operability" (Montgomery 2009). However, most of these designs do not include replicates, because they have been primarily used for deterministic computer models (Montgomery 2009). Another common type of designs are grid designs, also known as

factorial designs. These can be used to characterize complex response surfaces, though they are limited to few factors, as their size grows exponentially as the number of factors increases (Kleijnen et al. 2005).

A possible approach to align two models would be to run a space-filling experiment design for each model, and then compare the results produced by them. However, there is a significant drawback because we are interested in using stochastic, not deterministic, simulation models. Space-filling designs have many design points, so that the model must be run many times, even without replicates. Hence, the computational burden of running an experiment on a stochastic model would be greatly magnified when using such designs, and would, in addition, be doubled if we are comparing two models. Furthermore, from an information gain point of view, if the differences between models are few and very localized, many of those design points will yield little, if any, new information about the differences between models. A more efficient approach is needed which conserves computational effort while gaining valuable information. The space-filling design approach to model alignment can be thought of as a "brute-force" approach. The approach we propose, on the other hand, seeks to explore the differences in response surfaces intelligently, by detecting areas where information gain might be maximized and focusing on those.

Our approach was inspired partly by the response surface methodology's (RSM) iterative use of optimization and design of experiments. However, there are significant differences between both methods. RSM is used for determining the values of the input parameters that yield an optimal response. It is a sequential procedure that works as follows: a first-order model is fit as the result of a designed experiment. The path of steepest ascent (or descent, if it is a minimization problem) is found in which the response of the fitted model increases most rapidly. At each step along the path of steepest ascent, experiments are run until improvement ceases. At this point, a new experiment is run, to fit a new first-order model. When a first-order model no longer fits the true response surface, it means that the true surface has a curvature, which might indicate proximity to the optimum. At that moment, we fit a second-order model instead, and use it to approximate the location of the optimum point. Our approach shares the sequential nature of RSM, as well as its use of designed experiments to provide guidance to the optimization. That is the extent of their similarities. Our approach does not seek the input parameter values that optimize the response, but those that maximize the difference in the model responses. For a more in-depth presentation of RSM, see Montgomery (2009). That said, the use of first and second-order models to approximate the response surfaces, and using them in the optimization, is an idea that might be incorporated in our approach and could reduce the computational burden.

3 THE GENERALIZED MODEL ALIGNMENT FRAMEWORK

Our framework makes use of a two-phase methodology for model alignment. The first phase is an optimization (search) phase that explores the space and finds areas of interest, that is, it returns points in the space where it is most likely that the models differ. Once one of these areas has been found, the second phase performs a statistically designed experiment over the region of interest. This phase provides a statistical analysis of the outcomes of both models to determine whether or not they are equal. Furthermore, since the experiment is designed using the principles of statistical design of experiments, the results are collected to allow us to characterize the differences (if any). Once a conclusion is reached, phase I is repeated, but the region of interest that has already been explored is blocked out, preventing it from being explored again. This continues for as long as the user requires.

The following subsections discuss each phase in greater detail.

3.1 Phase I: Optimization

Suppose there are two simulation models denoted f_1 and f_2 respectively, which need to be aligned. Since we know nothing beforehand about the form of the response surfaces, we want to look for those areas where we have the greatest probability of detecting any differences between the two surfaces. We call

these "areas of interest." Figure 1 illustrates what we mean by areas of interest for two models with a single parameter. In the figure, the arrows show the two points (A and C) where the difference between the model responses is maximal, hence, those are the areas where an experiment is most likely to reveal that the two response surfaces are different. Point B, on the other hand, would require a more highly sensitive experiment to detect differences. Therefore, to get the most benefit with the least computational cost, it is better to evaluate the models at points A and C, and not at B.

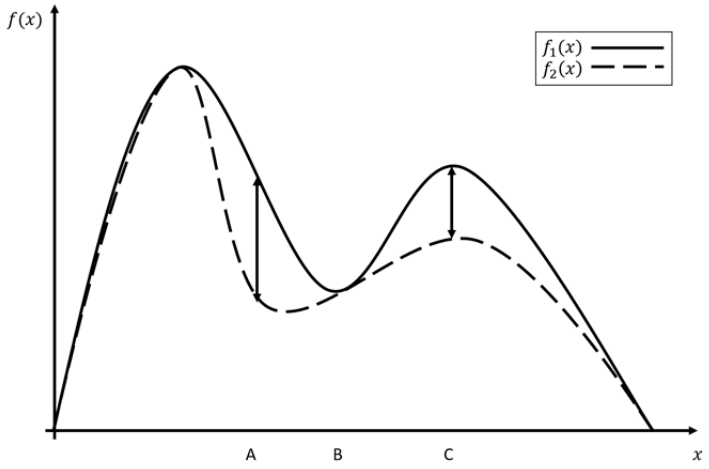

Figure 1: Response surfaces for two stochastic simulation models.

To detect these areas of interest, we rely on the following optimization model:

$$\max |f_1(\boldsymbol{x}) - f_2(\boldsymbol{x})|$$

Subject to a series of constraints on the possible values of the vector of parameters $\boldsymbol{x} \in \mathbb{R}^N$. For instance, the value of x_1 can be restricted to lie within the range a_1 to b_1 where the values of a_1 and b_1 vary as we explore different regions of the parameter space, allowing us to focus the search on those regions of the parameter space that have not been explored.

Since the simulation models f_1 and f_2 are stochastic, it is necessary to adjust our objective function to account for the variation in outcome that results from their stochastic nature. Hence, our objective function becomes:

$$\max |E[f_1(x)] - E[f_2(x)]|$$

Where $E[f_1(x)]$ and $E[f_2(x)]$ are the expected values of the responses for models f_1 and f_2 respectively, for a small sample of size n. The sample size will be dependent on the computational effort required by the simulation models, and by the desired accuracy in detecting areas of interest. Note that we use a linear error objective here but a quadratic or other error metric could be used equally easily.

3.2 Phase II: Designed Experiment

Once the points of interest are obtained by the optimization phase we perform a designed experiment on the region surrounding each point. For each experimental point (or parameter combination), we obtain a sample of observations from each model. Using the Kolmogorov-Smirnov test (and/or other tests of goodness of fit, such as the Chi-Squared or the Anderson-Darling tests), we can determine if these samples are drawn from the same distribution (effectively verifying their equality) or not. Additional tests

can help determine whether the differences are in the expected value (t-test for equality of means), the variance (Chi-squared test for equality of variance for two samples), or whether there is bias in one of the models, and so on.

Suppose the optimization phase finds, as we mentioned above, two points of interest, point A and point C. We can then perform a designed experiment using these points as center points. Figure 2 shows this for point A. In this case, we use a factorial design centered on point A, and sample from both models. The figure shows the results of taking a sample of size 3 from each experimental point.

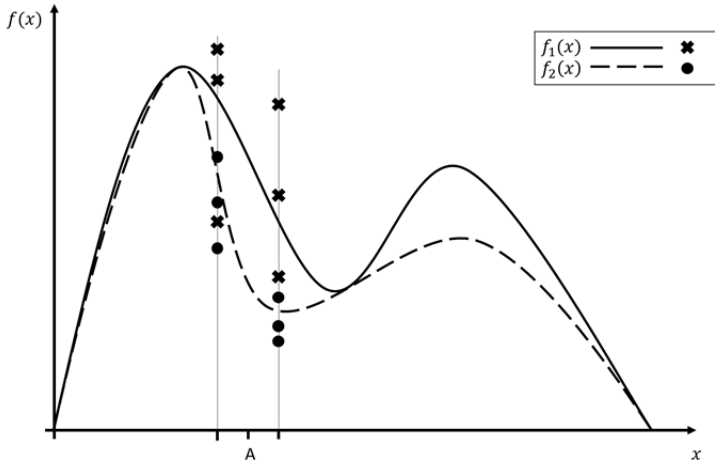

Figure 2: Experiment around point A.

The advantage of using a designed experiment is that it can provide additional information about how the two surfaces might be different. For example, interactions between parameters can be detected. If the designed experiment reveals an interaction between parameters for one model but not for the other, we conclude that there is a difference between the two models and know something about the nature of that difference. This discovery can help us discern why the two models are different. In the context of model transformation, it might bring to light some error in the transformation procedure.

4 A BRIEF EXAMPLE

To clarify the process we just described, consider this simple example. We want to compare two stochastic simulation models. The first model is implemented in one simulation platform, Simulink, while the second one is obtained from a transformation procedure that converts the model from a Simulink model to a RePast model. We want to know whether both models, despite running on different platforms, are the same. The models have three parameters: $\mathbf{x} = [x_1, x_2, x_3]$.

First, an optimization algorithm that seeks to find a point of maximal difference in responses is run. The set of best points found at each iteration of the algorithm are shown in Table 1.

Table 1: Results of the optimization algorithm.

Iteration	x_1	x_2	x_3	$f_1(\mathbf{x})$	$f_2(\mathbf{x})$	$\lvert f_1 - f_2 \rvert$
1	1.5	2.4	3.2	4.5	4.7	0.2
2	1.5	2.6	3.3	4.5	4.75	0.25
…	…	…	…	…	…	…
1000	5.6	6.5	4	1.7	6	5.7

The point of interest, then, is (5.6, 6.5, 4). This becomes the point around which a designed experiment is performed, a three factor with two levels per factor in a full factorial design. This is shown in both the encoded and decoded representation in Table 2. The user can determine what radius should be used to define the experiment around the central point. In this case, the radius is chosen to be 0.1, so that the low value of x_1 is 5.5, its high value is 5.7, and so on. The user can also specify the number of replicates per factor level combination.

Table 2: Encoded and decoded factor level combinations.

Encoded			Decoded		
x_1	x_2	x_3	x_1	x_2	x_3
0	0	0	5.5	6.4	3.9
0	0	1	5.5	6.4	4.1
0	1	0	5.5	6.6	3.9
0	1	1	5.5	6.6	4.1
1	0	0	5.7	6.4	3.9
1	0	1	5.7	6.4	4.1
1	1	0	5.7	6.6	3.9
1	1	1	5.7	6.6	4.1

The experiment is conducted by running the simulation model by using each factor level combination as the input parameter vector, and repeating each run for the specified number of replicates. The number of replicates is an important value because it must be large enough to make conclusions about each mode at each parameter combination to increase the power of the statistical tests. With the outcome of the experiment, we can already answer a few questions about the two models: are there differences in the significance of each parameter? Are there differences in the interactions between factors, or in the direction in which these affect the outcome (positively in one model, negatively in the other)?

Besides the comparison based on the analysis of the experiment, we compare each sample using the Kolmogorov-Smirnov test to determine, at a certain level of confidence, whether the samples are drawn from the same distribution or not. Other tests that can be used for similar analyses include the Chi-Squared test and the Anderson-Darling test.

We might also collect additional measures (expected value, variance, skewness, etc.) that can provide more information about how the models behave, and how they might potentially differ, even in the cases where the statistical tests reveal no difference. For instance, while the Kolmogorov-Smirnov test might fail to detect a difference in distribution between both models, we might detect that one model's response is consistently larger than the other, bringing to light a bias. The Kolmogorov-Smirnov test is based on the maximum vertical difference between the empirical cumulative functions of two samples (Darling 1957), so a small, yet consistent bias, might escape detection.

Once these analyses are completed for the region surrounding point A, the optimization phase resumes with additional constraints to keep the search away from point A, ensuring exploration of other regions of the parameter space. When a new region of interest is found, the designed experiment phase begins once more and the whole process is repeated.

5 PROOF OF CONCEPT DEMO OF THE MODEL ALIGNMENT METHOD

To test the feasibility of this approach, we developed a proof of concept demo using a quadcopter simulation model built in Matlab and available at https://github.com/gibiansky/experiments/tree/master/quadcopter under a CC BY-SA 2.0 license (Creative Commons 2017). The model was modified to incorporate some stochasticity (added as a random disturbance in the position of the quadcopter) and to produce a single output measure. The quadcopter model can use different control modules. We wanted to

test whether changing the control module results in different behavior for the quadcopter. The two controllers used were a Proportional-Integral (PI) controller, and a Proportional-Integral-Derivative (PID) controller. The controllers receive a set of tuning parameters as input. The PI Control has two parameters: the proportional (P) and integral (I) terms; while the PID controller has three parameters: a proportional, integral, and derivate (D) term. The model response is the distance traveled by the quadcopter, and the input parameters used are the proportional and integral terms required by the control module. For the PID controller, the derivative term was left at its default value.

For the optimization phase, we use a particle swarm metaheuristic that runs for 100 iterations with a swarm size of 10 particles. At each iteration, every particle runs both models 10 times (this sample size can be adjusted by the user) and measures the difference between the model outcomes. For the experimentation phase, we use a 2^2 full-factorial design, with the radius size around the central point predefined by the user. The number of replicates for each factor level combination was set to 30. All of these values can be modified by the user. A screenshot of the demo is seen in Figure 3.

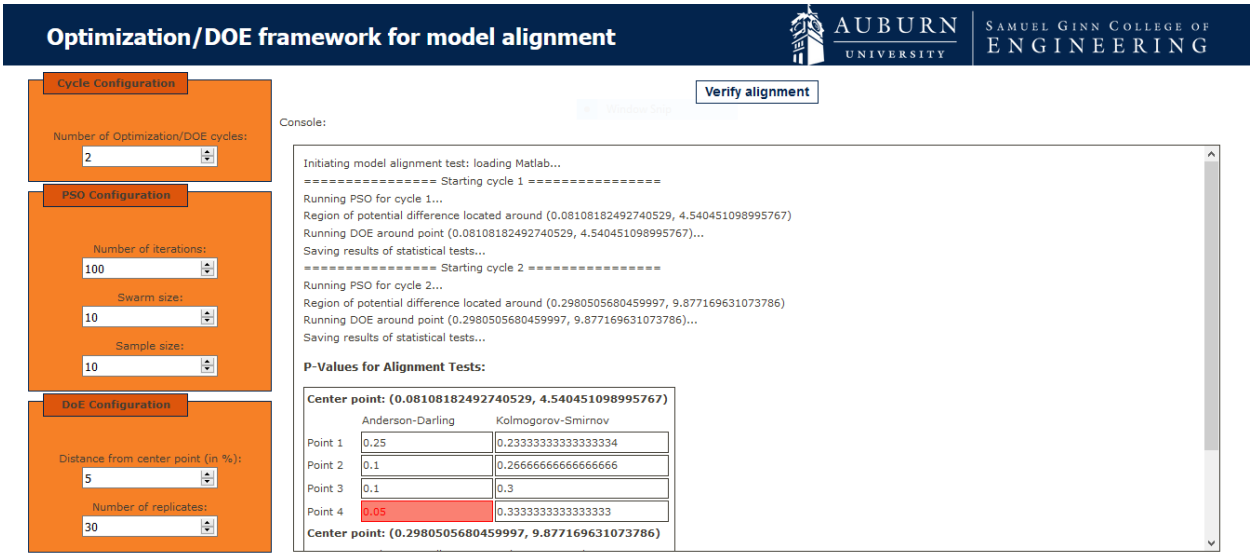

Figure 3: Demo screenshot.

The demo works as expected, scanning the response surfaces for differences, and evaluating regions of interest in greater detail. Using both a Kolmogorov-Smirnov and an Anderson-Darling test, it evaluates whether the samples from each model are drawn from the same distribution. In the figure, the cell highlighted in red indicates that during the evaluation of the region surrounding the first point of interest (obtained from the first run of the PSO), a difference was detected by the Anderson-Darling test. This test is more sensitive than the Kolmogorov-Smirnov test to differences in the tails of the distribution.

To better understand the difference between the proposed approach and the more traditional DoE methods, consider a 2 factor full-factorial design. A configuration of the proposed approach such as the one described above would result in 10,120 model comparisons (meaning twice as many model runs) per cycle. If we run for 10 cycles, we would have 101,200 comparisons. An approximately equivalent full-factorial would be a 58 factor level design, with 30 replicates per factor level combination, which results in 101,124 comparisons. Both approaches evaluate the simulation models roughly the same number of times, but the areas they explore are different. The full factorial evaluates the response surfaces evenly, while our approach focuses its search on certain areas. The full factorial design would compare the models in 58^2 = 3,364 different points. The proposed approach would explore, during the optimization phase, 10*100 = 1,000 points, plus 4 points from the DoE phase, per cycle, for a grand total of 10,040

different points. That said, the points explored during the optimization phase were not explored with the same level of detail as those evaluated during the DoE phase. In the optimization phase, each point was replicated 10 times, whereas during the DoE phase, they were replicated 30 times, for better precision in the comparison. Nonetheless, even the reduced sample size used in the optimization phase can produce information about the surfaces that might be of interest. The differences in exploration approaches can be seen, in a simplified manner, in the following figures.

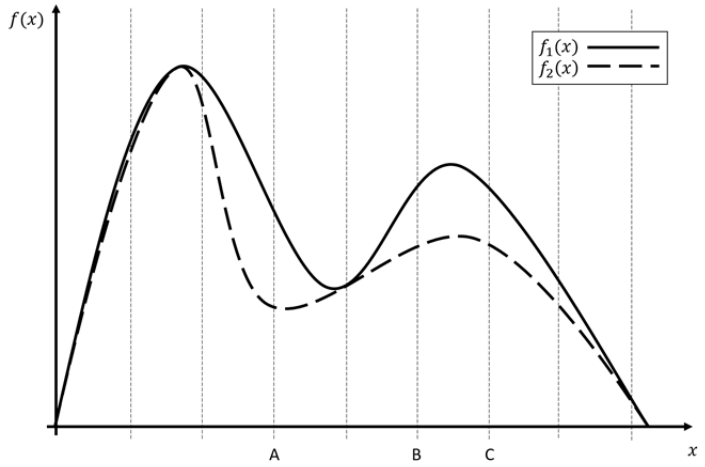

Figure 4: Full-factorial design.

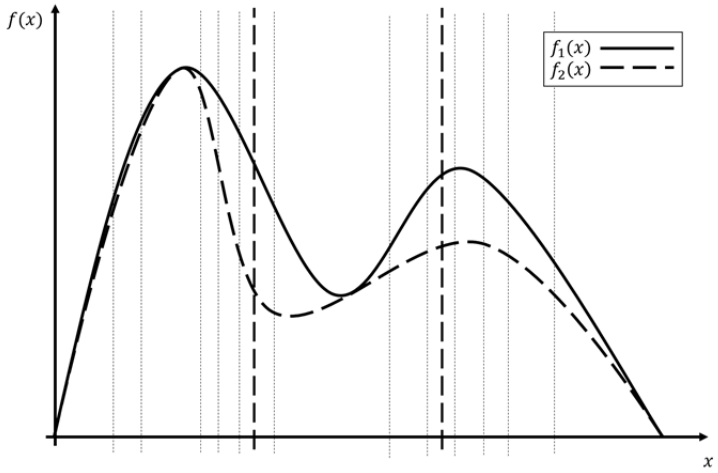

Figure 5: Optimization/DoE approach.

The full-factorial design (or any space-filling design) is a better approach if what is intended is to characterize the response surface, since it offers an unbiased sampling of the entire surface. The even distribution of experimental points allows for the observation of the surface over its full range. However, if what is sought is to find differences between response surfaces, then many observations in such designs will yield little information. In Figure 4, only points A, B, and C are useful for detecting the difference in the two surfaces. The only information provided by the remaining observation points is that the two models are statistically identical. In Figure 5, on the other hand, even those observation points used in the optimization phase (indicated with a thin dotted line), provide some information about the location of points where the probability of detecting a difference is greater, and it is these points that are evaluated in

greater detail during the DoE phase (indicated in the figure by a thick dashed line). While in the full-factorial, 10 points are evaluated in detail, of which 7 yield little information, in the proposed approach, only 2 points are evaluated in detail, and the additional points help us locate them. The proposed approach is a biased search, because its intent is not to explore the entire surfaces, but only those parts of them that are of interest.

This behavior is also observed when we compare the outcomes of both approaches. The full-factorial approach yielded, as was mentioned above, 3364 observation points, of which 138 indicated there was a difference between the models. An analysis of these revealed that there were several clusters, or regions where the differences were concentrated. 10 different regions were found. A region was defined if there were at least 4 observation points in close proximity that signaled a difference between models. For example, Region 1, ranging roughly from point (0, 3) to (0, 5), consists of 11 points, of which 10 indicated a difference between the models. At the same time, there were several points where a difference was detected but which do not fall into any of the regions defined. The proposed approach evaluated 40 observation points in detail, of which 4 detected differences. The remainder of the points, however, were located in or close to the regions found by the full-factorial, as can be seen in Table 3.

Table 3: Location of proposed approach observations vs. regions found by full-factorial.

Center point found by optimization	In or near region
(0.1589, 3.777)	1
(1.5568, 0)	None
(0.5177, 10)	2
(0.6678, 9.9394)	2
(0.1376, 10)	2
(0.1739, 2.8040)	1
(1.3217, 9.1310) *	None *
(0.1375, 0)	2
(0.1636, 6.4042)	2
(0.3525, 7.3217)	2

The optimization/DoE approach indicated a difference between models in the neighborhood of the point marked with an asterisk. Though this point is not near or in a region of those identified by the full-factorial experiment, the latter identified an isolated observation point where there was a difference fairly close to it (1.4035, 8.5964). What these results show is that the proposed approach is focused on exploring those areas where differences are concentrated (in particular regions 1 and 2). The fact that it does not explore other areas of interest (the remaining 8 regions identified by the full-factorial) is due to the optimization algorithm finding the same optima at each cycle. This can be addressed by using the results of previous iterations and using them to restrict the search space, so to direct it in new directions.

The proposed approach has an additional advantage with respect to the full-factorial. The power of the comparison test can be improved by running more replicates per observation point, that is, by using a larger sample. Running one more replicate per point would imply 3,364 additional comparisons for the full-factorial experiment. For the proposed approach, that would imply only 40 additional comparisons. As a result, we can more readily use more replicates with the proposed approach, which in turn would improve the precision of the test.

6 CONCLUSIONS AND FUTURE WORK

This paper discussed an approach to model alignment for detecting differences in the responses of two simulation models. The context of this work is that of model transformation for experiment replicability.

A proof of concept demo is developed that shows how our approach can work, though the model used in it is fairly simple and runs very quickly.

We foresee the following obstacles to using our proposed approach on larger scale models. The most pressing obstacle has to do with the computational burden or execution time of the models. This approach requires multiple runs of the models being compared, both for the optimization and the design of experiment phases. Many optimization procedures, particularly population based meta-heuristics like the particle swarm used in our demo, rely heavily on repeated evaluations of the objective function. Hence, this approach (at least in its current form) will have difficulties dealing with slow or burdensome simulation models. On the other hand, the approach is not bound to a specific optimization algorithm, so that one that requires less model evaluations could be used instead. Another obstacle has to do with the size of the parameter space. As the number of model parameters multiplies, as does their possible range of values, the space size grows exponentially. In such a case, even an effective optimization procedure would need a long time to find areas of interest. This approach can be used to explore as much of the parameter space as possible, but it can also be adjusted to explore only parts of it that might be of interest to the researcher.

Though the motivation for developing this method arose from work in model transformation for scientific replicability and reproducibility, it can have applications in other areas of modeling and simulation. It might, for instance, be used to test the validity of simulation metamodels, or, as Edmonds and Hales (2003) mention, to detect errors in implementations or reveal hidden assumptions. It can also be used to compare the simulation model to data from the real-world system, if such is available.

Our future work is aimed at addressing the major obstacles mentioned above, which includes testing our demo with larger scale models and analyzing the effects of using different optimization algorithms. We also need to further investigate how different designed experiments can yield more valuable information to characterize the differences between the models, or how the meta-models developed from a designed experiment might reduce the computational burden during the optimization phase.

Another important area of future work is one looking into the statistical methods used by our approach. The demo is based on a comparison of expected values, but this is clearly not sufficient. Comparisons aimed at detecting bias, differences in the variance of both models, among others, are also important. These comparisons would require different statistical tests than the ones discussed in this paper. Lastly, it is also important to define methods for specifying the sample sizes to be used both in the optimization and the design of experiment phases. Besides statistical considerations, one must also take into account the computational effort required by the model, as well as the acceptable accuracy of the comparison. Tests concerning the robustness of different sample sizes would be particularly interesting.

ACKNOWLEDGEMENTS

This research was sponsored by an unnamed agency of the U.S. Federal Government.

REFERENCES

Axtell, R., Robert A., J.M. Epstein, and M.D. Cohen. 1996. “Aligning Simulation Models: A Case Study and Results.” *Computational & Mathematical Organization Theory* 1 (2): 123–141.

Creative Commons. 2017. “Attribution-Share Alike 2.0 Generic License.” Accessed April 25. https://creativecommons.org/licenses/by-sa/2.0/legalcode.

Darling, D.A. 1957. “The Kolmogorov-Smirnov, Cramer-Von Mises Tests.” *The Annals of Mathematical Statistics* 28 (4): 823–838.

Edmonds, B., and D. Hales. 2003. “Replication, Replication and Replication: Some Hard Lessons from Model Alignment.” *Journal of Artificial Societies and Social Simulation* 6 (4).

Fachada, N., V.V. Lopes, R.C. Martins, and A.C. Rosa. 2016. “Parallelization Strategies for Spatial Agent-Based Models.” *International Journal of Parallel Programming*, 1–33.

Fachada, N., V.V. Lopes, R.C. Martins, and A.C. Rosa. 2017. "Model-Independent Comparison of Simulation Output." *Simulation Modelling Practice and Theory,* 72: 131–49.

Kleijnen, J.P.C., S.M. Sanchez, T.W. Lucas, and T.M. Cioppa. 2005. "State-of-the-Art Review: A User's Guide to the Brave New World of Designing Simulation Experiments." *INFORMS Journal on Computing* 17 (3): 263–289.

McKay, M.D., R.J. Beckman, and W.J. Conover. 1979. "Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code." *Technometrics* 21 (2): 239–245.

Miodownik, D., B. Cartrite, and R. Bhavnani. 2010. "Between Replication and docking:' Adaptive Agents, Political Institutions, and Civic Traditions' revisited." *Journal of Artificial Societies and Social Simulation* 13 (3): 1.

Montgomery, D.C. 2009. *Design and Analysis of Experiments*. Vol. 7. Wiley New York.

Pronzato, L., and W.G. Müller. 2012. "Design of Computer Experiments: Space Filling and beyond." *Statistics and Computing* 22 (3): 681–701.

Wilensky, U., and W. Rand. 2007. "Making Models Match: Replicating an Agent-Based Model." *Journal of Artificial Societies and Social Simulation* 10 (4): 2.

Will, O. 2009. "Resolving a Replication That Failed: News on the Macy & Sato Model." *Journal of Artificial Societies and Social Simulation* 12 (4): 11.

Will, O., and R. Hegselmann. 2008. "A Replication That Failed on the Computational Model in'Michael W. Macy and Yoshimichi Sato: Trust, Cooperation and Market Formation in the US and Japan. Proceedings of the National Academy of Sciences, May 2002'." *Journal of Artificial Societies and Social Simulation* 11 (3): 3.

Xu, J., Y. Gao, and G. Madey. 2003. "A Docking Experiment: Swarm and Repast for Social Network Modeling." In *Seventh Annual Swarm Researchers Meeting (Swarm2003)*, 1–9.

AUTHOR BIOGRAPHIES

ALEJANDRO TERAN-SOMOHANO is a Ph.D. Candidate in Industrial and Systems Engineering at Auburn University. He holds a Bachelor's degree in Computer Engineering from the Instituto Tecnologico Autonomo de Mexico (ITAM) and an M.S. degree in Industrial and Systems Engineering from Auburn University. His email address is ateran@auburn.edu.

ALICE E. SMITH is the Joe W. Forehand / Accenture Distinguished Professor of Industrial and Systems Engineering at Auburn University with a joint appointment in Computer Science and Software Engineering. She has authored papers with over $2,000 ISI Web of Science citations and has been a principal investigator on projects with funding totaling over $6 million. She is an area editor of *INFORMS Journal on Computing* and *Computers & Operations Research* and an associate editor of IEEE Transactions on Evolutionary Computation and *IEEE Transactions on Automation Science and Engineering*. Her email address is smithae@auburn.edu.

LEVENT YILMAZ is Professor of Computer Science and Software Engineering at Auburn University with a joint appointment in Industrial and Systems Engineering. He holds M.S. and Ph.D. degrees in Computer Science from Virginia Tech. His research interests are in agent-directed simulation, cognitive computing, and model-driven science and engineering for complex adaptive systems. He is the former Editor-in-Chief of *Simulation: Transactions of the Society for Modeling and Simulation International* and the founding organizer and general chair of the Agent–Directed Simulation Conference series. His email address is yilmaz@auburn.edu.