Proceedings of the 2021 Winter Simulation Conference
S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, eds.

SIMULATION OPTIMIZATION FOR A DIGITAL TWIN USING A MULTI-FIDELITY FRAMEWORK

Yiyun Cao
Christine Currie
Bhakti Stephan Onggo

Centre for Operational Research,
Management Science and Information Systems
University of Southampton
Southampton, SO17 1BJ
United Kingdom

Michael Higgins

PowerTrain Manufacturing Engineering
Ford Motor Company
Dunton Research Centre
Basildon, SS15 6EE
United Kingdom

Abstract

Digital twin technology is increasingly ubiquitous in manufacturing and there is a need to increase the efficiency of optimization methods that use digital twins to answer questions about the real system. These methods typically support short-term operational decisions and, as a result, optimization methods need to return results in real or near-to-real time. This is especially challenging in manufacturing systems as the simulation models are typically large and complex. In this article, we describe an algorithm for a multi-fidelity model that uses a simpler low-fidelity neural network metamodel in the first stage of the optimization and a high-fidelity simulation model in the second stage. Initial experimentation suggests that it performs well.

1 Introduction

We consider the use of digital twin technology to optimize a system. Optimization via simulation has traditionally been used to find the best solution using a static simulation model. When optimization is being run with a digital twin simulation, there is a greater need for results to be produced quickly and efficiently as these models tend to be used to answer immediate operational questions. The methodology we describe here forms part of a larger project with Ford Motor Company and we include some discussion of this in what follows.

Digital twin models go hand in hand with advances in Industry 4.0 as they rely on data from sensors. The digital twin is a virtual replica of a real machine, system or process, which can be used to simulate its behavior (Negria et al. 2017) and they have mainly been used in design, operation and maintenance. The key difference between a digital twin and a standard simulation model is that a digital twin is synchronized using real-time data from the physical system. Symbiotic simulation systems describe the whole process of using a digital twin (Onggo et al. 2021; Onggo 2019), which are also known in the literature as dynamic data-driven simulation models or rolling horizon models. In a symbiotic simulation system, optimization and/or machine learning can be combined with the digital twin to suggest the best possible decisions given the limited time to make a decision.

Simulation has been an important tool in the design and management of Ford's manufacturing facilities since the early 1980s. Higgins and Ladbrook (2018) gives an overview of the modeling process that they use and its evolution over the past two decades. In the final section of the paper, Higgins and Ladbrook cite "Models that run themselves to align with current production" and symbiotic simulation as important future directions at Ford. The optimization method we develop here fits within that process, allowing fast

978-1-6654-3311-2/21/$31.00 ©2021 IEEE

experimentation with a complex simulation model and a large decision space. The real-time or near real-time simulation optimization lies at the core of Ford's strategy with digital twin or symbiotic simulations.

The simulation models that Ford uses to describe their production lines are large, detailed, complex and therefore, computationally expensive. In order to improve the efficiency of the simulation optimization and achieve the near-real-time results required for their operational decision-making, it is important to minimize the number of replications using these computationally expensive models. As a result, in this article, we consider *multi-fidelity simulation optimization*, where a simple or low-fidelity model is used in the first stage of the optimization to guide the solutions to test on the detailed or high-fidelity model in the second stage of the optimization. This article complements previous work in the area, which led to the development of MO^{2}TOS (Xu et al. 2014), but uses a different grouping strategy for results from the low-fidelity model. Results on a test case study suggest that it achieves good enough results with a relatively small computing budget.

The remainder of this article is organised as follows. We review related works in Section 2. Section 3 presents our algorithm, which we test on a case study in Section 4, before concluding and describing future work in Section 5.

2 LITERATURE REVIEW

We divide the literature review into three sections, beginning with a discussion of the literature associated with digital twins and symbiotic simulation before continuing to the discussion of simulation optimization and then the more focused topic of multi-fidelity simulation optimization.

2.1 Digital Twins and Symbiotic Simulation

As discussed in Section 1, the core concept of a digital twin is that a virtual replica of the real-world system runs alongside the real, physical system. When demand changes or the system updates, data are sent to the digital twin and parameters are reconfigured or the state of the model is modified. The concept has been prevalent in many different industries since the idea was mooted by NASA during the Apollo project in 1969 (Glaessgen and Stargel 2012). It also goes under different names: digital twins (Glaessgen and Stargel 2012), real-time simulation (Pedrielli et al. 2019), and symbiotic simulation (Onggo et al. 2021; Onggo 2019) being the most common. The aim of the technology is to enable real-time decision-making that accounts for the current state of the system.

Zhang and Zhu (2019) claim that digital twins have three key parts: object (or real-world system), model and data. While their focus is on its use at machine level, digital twin technology is applicable from machine level to production line level (manufacturing system) to factory level (system of systems). Most research lies in the optimization of production processes, fault diagnosis and prognostics. A digital twin will depend on data from the object being modeled and this may come from sensors or, as Ivanov et al. (2019) describes, it may make use of radio-frequency identification (RFID) and geographic information systems (GIS) to update the system state. The choice of inputs will, to a certain extent, be dependent on the nature of the system being modeled but also on the availability of data.

A symbiotic simulation system comprises the digital twin, optimization methods, modules for data acquisition and an actuator for system control, as well as the corresponding physical entity (Onggo et al. 2021). The original definition is given by Aydt et al. (2008) who defined symbiotic systems as those having "a close association between a simulation system and a physical system, which is beneficial to at least one of them." The association is typically the exchange of data from the real system to the digital twin and the communication of recommendations that emerge from the optimization to the real system. Typically, heuristics and machine learning are used for the optimization. One of the challenges is to enable simulation models to respond to the feedback from physical systems, which keeps models up to date.

2.2 Simulation Optimization

Simulation optimization or optimization via simulation methods use the simulation as a proxy for the real system and aim to maximize/minimize a given objective function. As the simulation models we work with (usually Discrete Event Simulation (DES) models) have stochastic or random output, the objective is typically the expected value of the model output of interest. Using mathematical notation, we wish to minimise an output $f(\mathbf{x})$, where $\mathbf{x}$ is a vector of decision variables and $f(\mathbf{x})$ is the expected value of the random output $Y(\mathbf{x})$,

$$f(\mathbf{x}) = \mathbb{E}[Y(\mathbf{x})].$$

We assume in what follows that we have a single objective problem and consequently the output $f(\mathbf{x})$ is a single number.

In this paper we focus on a problem where the feasible region for $\mathbf{x}$ has a finite number of discrete solutions. In the Ford example, these could represent different repair policies or different strategies for running the production line. Problems of this nature are described as *ranking and selection* problems and two main algorithms exist within the literature for solving them: indifference zone procedures such as KN++ (Kim and Nelson 2006) and optimal computing budget allocation or OCBA methods (Chen and Lee 2010). Indifference zone procedures will estimate the number of replications needed for each solution to identify the correct optimum with a given degree of certainty. OCBA and other budget allocation methods will instead optimally allocate a fixed computational budget between different solutions, typically in order to maximize the probability of correct selection. We make use of OCBA in what follows.

2.3 Multi-fidelity Simulation Optimization

Multi-fidelity simulation optimization methods make use of two models of the real system: a low fidelity model that is computationally fast to run but may include less of the system's complexity, and a high fidelity model that is generally slower to run. Exploratory analysis using the low-fidelity model can help to guide the second stage of optimization using the high fidelity model. In the situations we consider here, the high fidelity model is the DES model. Multi-fidelity optimization algorithms have been shown to be a feasible way to boost the efficiency of simulation optimization as shown in the following examples.

Multi-fidelity methods often extend original stochastic optimization methods to incorporate an initial exploration using a low-fidelity model. For example, Multi-Fidelity Sequential Kriging Optimization (Huang et al. 2006) extends Sequential Kriging Optimization (Huang et al. 2006), with the algorithm selecting the fidelity level and data points for the simulation in the next step to maximize the expected improvement. Value-based Global Optimization (Moore et al. 2014) has a similar structure using a utility function to extract information of value (IoV) rather than expected improvement to select the next point at which to refine the metamodel. The process ends when the IoV is no longer positive.

In work that is more similar to the approach we use here, Horng and Lin (2009) proposes a two-stage procedure to solve G/G/1/K polling system problems with K limited, where the aim is to select the optimal service limit. In the first stage, an Artificial Neural Network (ANN) assisted Genetic Algorithm (GA) downsizes the solution set for optimization. The second stage uses a short simulation run and a longer simulation run. The shorter run is carried out first to reduce the target area, and the longer run is then used to compute the optimum. In Horng and Lin (2013), the first stage uses a GA with a fitness function structured by an ANN, as in their 2009 work, to produce a smaller subset of alternatives but an OCBA procedure is applied in the second stage to select the best.

In many situations, the low-fidelity model is a deterministic model of the system. For example, in Rhodes-Leader et al. (2018), which aims to solve the aircraft-recovery problem, the authors use an integer program for the low-fidelity model. Results of the integer program are then evaluated using the simulation model. Osorio and Bierlaire (2013) combine a quadratic polynomial model, which they describe as a *functional metamodel*, with an analytical queueing network model to form their low fidelity model when optimizing a traffic network. Combining the quadratic polynomial model with a physical-based model that

incorporates some of the key features of the system reduces the amount of simulation time needed to fit the low-fidelity model.

Another group of work uses multi-fidelity methods to improve the accuracy of the metamodel. Han and Görtz (2012) proposed a hierarchical kriging model for a continuous problem, which maps the trend of the lower-fidelity model to the sampled high-fidelity data in order to apply any necessary corrections. Lin et al. (2019) combines simulation with a kernel regression metamodel. Their method generates accurate estimates rapidly by locally correcting the low-fidelity outputs through high-fidelity samples.

Recent research has suggested that all feasible solutions should be evaluated in the low-fidelity model and the preference order be input into the high-fidelity model. For example, Xu et al. (2014) propose the Multi-fidelity Optimization with Ordinal Transformation & Optimal Sampling (MO^{2}TOS) algorithm to leverage the low-fidelity model for narrowing down the design space. It places solutions with close performance in the low-fidelity model into one group by evenly splitting the solution space. The groups are then input into a ranking and selection (R&S) method using the high-fidelity simulation model for evaluation, where each group is treated as a single solution. Xu et al. (2016) use OCBA (Fu 2014) in MO^2TOS for the optimal sampling. When a group is assigned sampling budget, solutions in the group are sampled without replacement. The optimal solution is then picked from the sampled solutions. OCBA takes both the mean and variance of the candidates into consideration when allocating sampling budget in order to maximize the probability of correct selection. Following completion of OCBA, the best sampled solution is selected and returned. Typically, a simplified model is used for the low-fidelity model and a more complex simulation for the high-fidelity model. Song et al. (2019) introduced the Expected Optimality Gap (EOG) to MO^2TOS to measure the performance of optimal sampling and in MO-MO^{2}TOS, Li et al. (2016) extended MO^{2}TOS to work with Pareto fronts to solve multi-objective problems.

An alternative to MO^{2}TOS is the Generalized Ordinal Learning Framework (GOLF) (Pedrielli et al. 2019), which was proposed to support real-time decision making where data are generated by both the simulation model and the real system. Learning is carried out offline, as in our method, and GOLF partitions the design space into regions, allowing it to focus sampling on areas most important to the optimization.

3 METHODS

We use a multi-fidelity framework with a DES model as the high-fidelity model and a multi-layer feedforward ANN metamodel (Ojha et al. 2017) as the low-fidelity model. The left side of the flowchart in Figure 1 shows how we develop the metamodel. First, we train the metamodel using the high-fidelity simulation model to the output results of interest for a set of training data. Next, we validate the metamodel by comparing its estimates with output of the simulation model on a validation data set. If the metamodel is a poor fit, we generate more simulated data and refit the metamodel. The validated low-fidelity model is then used in the first step on the right side of the flowchart (i.e. we evaluate solutions via the metamodel).

The right side of the flowchart shows how our multi-fidelity framework works. First, we use the validated metamodel to evaluate the solutions. Next, we use the hierarchical agglomerative clustering method (Day and Edelsbrunner 1984) to partition the solutions into N clusters based on their performance, where N is predetermined. Then, the clusters are indexed in the order of their performance, where cluster 1 contains the most-promising solutions and cluster N contains the least-promising solutions. We put the clusters into groups for the next stage of the process, where the number of groups is predetermined. The grouping is designed so that the high-rank groups contain fewer but more-promising clusters and low-rank groups contain more but less-promising clusters. This contrasts with the grouping used in MO^{2}TOS (Xu et al. 2014), where solutions are split evenly between the different groups. By clustering first and then grouping we have more control over the number of solutions in each group. This allows us to have a smaller number of solutions in the high quality group and a larger number in the lower quality group. For example, in the case study in Section 4 we group 31 clusters into 5 groups with a preset proportion of 1:2:4:8:16. Clusters are arranged into groups as follows.

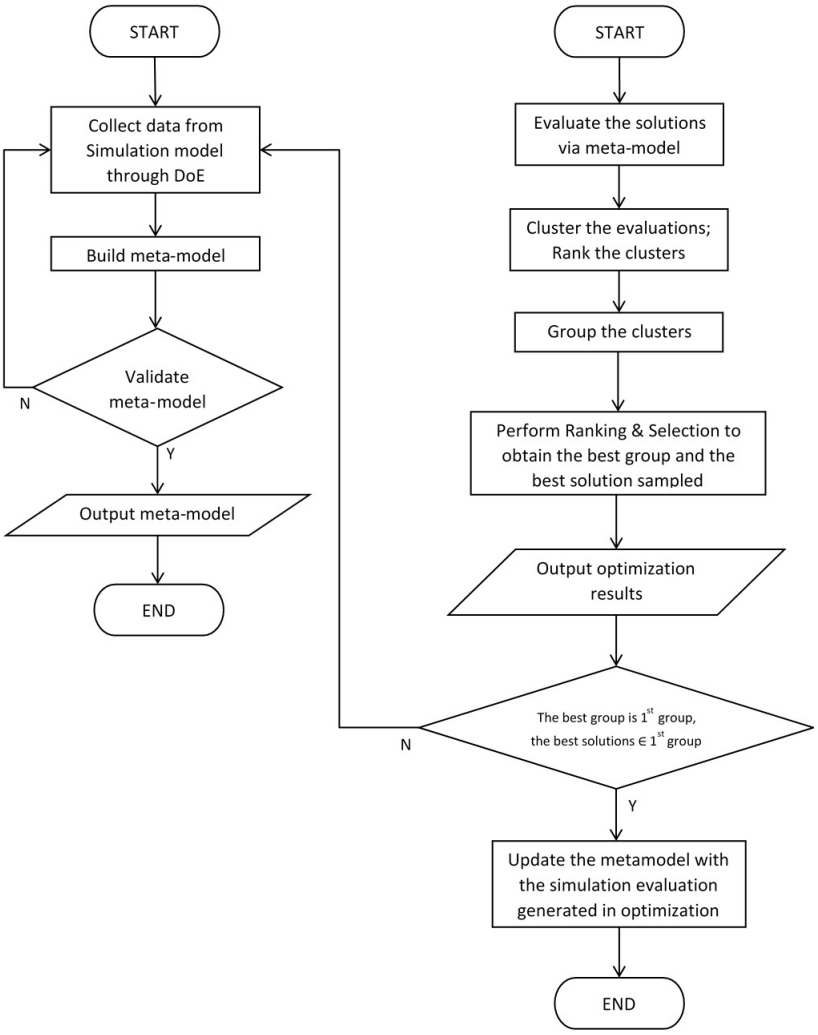

Figure 1: Flowchart of the proposed algorithm. Left: offline process, output fitted metamodel. Right: online optimization process, output final solution.

$$\text{Group } 1 = \{\text{Cluster } 1\}$$

$$\text{Group } 2 = \{\text{Cluster } 2,\ \text{Cluster } 3\}$$

$$\text{Group } 3 = \{\text{Cluster } 4,\ \text{Cluster } 5,\ ...,\text{Cluster } 7\}$$

$$\text{Group } 4 = \{\text{Cluster } 8,\ \text{Cluster } 9,\ ...\ \text{Cluster } 15\}$$

$$\text{Group } 5 = \{\text{Cluster } 16,\ \text{Cluster } 17,\ ...\ \text{Cluster } 31\}$$

In the final stage of the optimization, we follow the same procedure as MO^{2}TOS, and use a R&S procedure to allocated sampling budget to each group before sampling a solution from the chosen group and using this as the input to the simulation model. There is no guarantee that all solutions will be sampled at this stage but our method of grouping means that there is a higher chance of the best solutions being sampled and does not preclude any solution from being sampled. At the end of the R&S procedure, the optimization outputs the five solutions with smallest (or largest) estimates by the high-fidelity model. Future

work will explore the possibility of applying one more round of R&S for the selected best group, balancing improvements in the final solution with an increase in computation time.

Finally, we apply the following two questions to the best solutions obtained by the R&S procedure:

1. Is the best group selected by the algorithm the same as the first group?
2. Do the best solutions belong to the first group?

A negative response to either of these questions suggests that the metamodel needs to be refitted before the next call to the online optimization. Additional replications will be run for solutions with the greatest discrepancies between the metamodel and the simulation model.

4 CASE STUDY

We use an inventory system model from Law (2015) to demonstrate our algorithm. This is a well-known example that uses a DES model to describe the system behavior and the effect of ordering policies on costs. Ordering policies are described by two decision variables, (s,S), where s is the reorder point of the inventory system and S is the maximum inventory. At each timestep, the ordering policy suggests an order size of Z where

$$Z = \begin{cases} S-I & \text{if } I < s \\ 0 & \text{if } I \geq s \end{cases}$$

and I is the inventory level at the beginning of the month. The optimal ordering policy minimizes the average total cost, i.e. the sum of the order cost, holding cost and shortage cost. We constrain the range of decision variables such that $s \in [1,150]$, $S \in [2,151]$ and $s < S$. This results in 11,325 feasible solutions.

4.1 Building the Metamodel

This section shows the results from the metamodel development, as described on the left side of the flowchart in Figure 1. The metamodel is fitted offline and consequently there is less of a time constraint on this procedure but we still assume that the computational time should not be excessive. As a result we cannot run the simulation model at every single design point. In future iterations of this algorithm, this procedure may become more time-critical if we are updating the model and the metamodel in real-time.

The neural network metamodel was structured with Keras (Chollet, F. 2015), where we use the mean absolute error (MAE) for the loss function, Adaptive Moment Estimation (ADAM) (Kingma and Ba 2015) for the optimizer and Gaussian Error Linear Units (GELU) (Hendrycks and Gimpel 2016) for activation functions. For simplicity, the neural network is designed to have the same number of neurons in its hidden layers. The number of neurons and hidden layers were tuned to reduce the loss, resulting in a model with 9 hidden layers, each with 50 neurons.

We use Latin Hypercube Sampling (Morris and Mitchell 1995), a space-filling experimental design, to generate design points for training, testing and validating the metamodel. The same number of replications of the high-fidelity simulation model are run at each design point. We assume a fixed computational budget for the experimentation and consider two options for the sampling: a single replication at each design point or multiple replications at fewer design points. Specifically we set the budget to 100 and consider running the simulation once at each of 100 design points (100×1) and 5 times at each of 20 design points (20×5). We test the quality of the metamodel by comparing it with actual results from the simulation model and find that the MAE of the (20×5) strategy is 5.62, while the MAE for the (100×1) strategy is 2.06. As a result we use the metamodel fitted using the (100×1) strategy in what follows.

A three-dimensional surface plot showing the fitted metamodel is presented in Figure 2. This shows that estimated costs are high when s and S are both at their extremes.

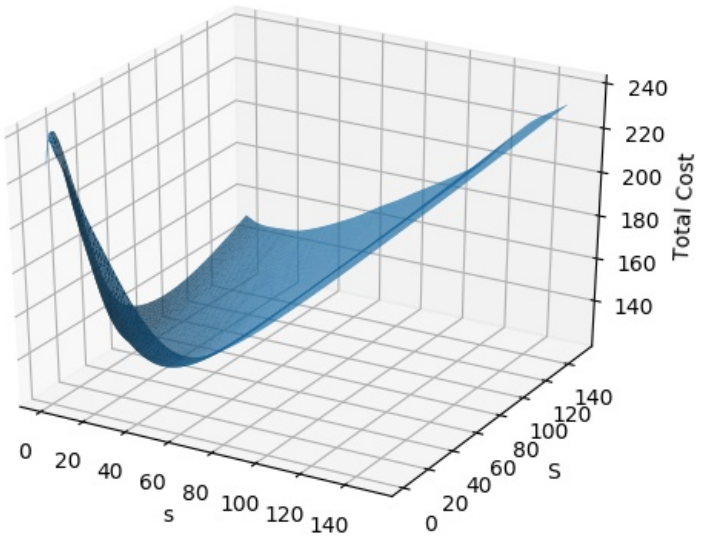

Figure 2: Evaluation of solutions from metamodel

4.2 Optimization

This section shows the results from the multi-fidelity framework experiments as described on the right side of the flowchart in Figure 1. In Figure 3 we plot the metamodel estimates of the total cost for each feasible solution (blue line), where solutions have been placed in a non-decreasing order. The results of hierarchical clustering of these estimates are superimposed via orange vertical lines and group membership is also indicated in the figure. The x-axis is the ranking of candidate policies and the y-axis is the average total cost calculated from the metamodel. We can see from the plot that data points are closer together for higher values of the total cost.

We dictate the number of clusters to be 31 to ensure that we have 5 groups at the end of the process. The grouping strategy merges the clusters following the preset proportions (1:2:4:8:16). The number of solutions contained in each group is given in Table 1 and the groups are shown visually in Figure 4. The first group contains the fewest solutions (602) and the fourth group the most (3856), although the fifth group accommodates the most clusters.

Table 1: Structure of the groups

Group	No. of solutions	No. of clusters
1	602	1
2	1205	2
3	2506	4
4	3856	8
5	3156	16

The five groups described in Table 1 are treated as five candidate solutions by the OCBA algorithm, which has its maximum number of replications set to 50. This step equates to online sampling and consequently is time-critical, hence the small number of replications for the OCBA. As detailed in Section

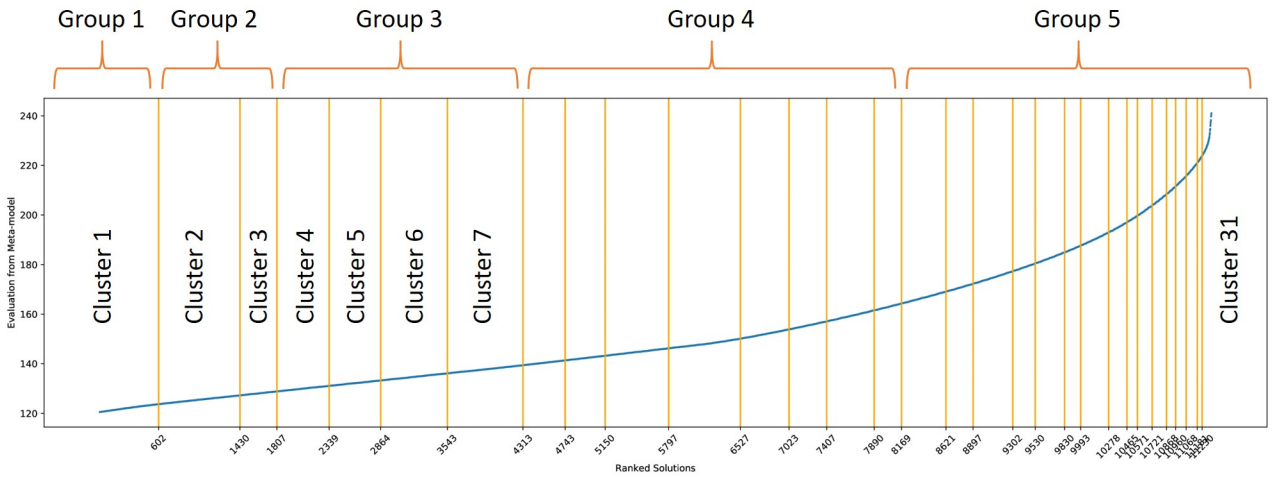

Figure 3: Clustering on ranked solutions

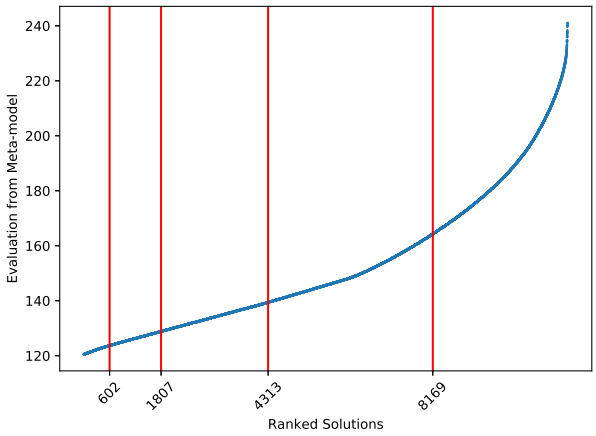

Figure 4: Grouping clusters of ranked solutions

3, when OCBA selects a particular group for sampling, the first step is to choose (at random) which solution will be selected from the group and the second step is to run the DES model with this chosen solution.

4.3 Results Analysis

To provide a comparison, we run 5 replications of the DES at each feasible solution and record the results and label this as the Uniform Allocation (UA) result. Table 2 shows the top five solutions identified by our algorithm for a single run of the optimization algorithm, their rank under the UA policy and the costs and regret estimates from the UA policy with 90% confidence intervals. Here the regret is defined to be the difference between the total cost estimate of the chosen solution and the total cost estimate of the optimal solution, as identified by UA. The optimal solution identified by the UA is $(23, 69)$, with an average total cost of 118.953 ± 1.674 with 90% confidence interval. The confidence intervals for the 'Regret' are calculated using 90% paired-t confidence intervals. We use common random numbers in the experiments to reduce variability.

We see from the results that the estimated mean regret is less than 3% of the best for all five solutions, with the smallest regret being 0.7%. Policy $(35, 70)$, which ranks 5th is the only solution in the top five

that is significantly different from the best. All five solutions come from the first group, which was selected as the best solution in the OCBA algorithm.

Table 2: Top 5 solutions from proposed optimization

No.	(s,S)	Rank in *UA*	Evaluation in *UA*	Regret from the optimum (*UA*)
1	(26, 66)	112	119.783±1.861	0.83±2.184
2	(23, 64)	140	119.89±2.379	0.937±2.537
3	(21, 60)	244	120.431±2.055	1.478±2.311
4	(25, 74)	221	120.295±0.992	1.342±1.697
5	(35, 70)	631	122.293±1.553	3.34±1.991

We also compared our algorithm with MO^2TOS (Xu et al. 2016). The results of applying MO^2TOS to this example can be found in Table 3. Two policies out of the top five ((20,90) and(12,86)) are significantly different from the best identified by the UA and the largest mean regret of these top five solutions is close to 7%, higher than that observed using our algorithm.

Table 3: Top 5 solutions from MO^2TOS

No.	(s,S)	Rank in *UA*	Evaluation in *UA*	Regret from the optimum (*UA*)
1	(24, 62)	109	119.778±1.897	0.825±2.207
2	(22, 54)	335	120.805±2.877	1.852±2.903
3	(20, 90)	1081	124.655±1.561	5.702±1.996
4	(24, 51)	128	119.838±2.029	0.885±2.295
5	(12, 86)	1660	127.139±1.733	8.186±2.101

A single run of the optimisation cannot provide a conclusive result because both the optimisation algorithm and DES models involve multiple stochastic processes. Therefore, 100 experiments were carried out for both our method and MO^{2}TOS. One experiment here refers to one optimisation run. We count how many results in the top-5 solutions are not significantly different from the optimum for each experiment. The histogram of the results is given in Figure 5 and shows that for our proposed method, in 50% of experiments all of the top-5 solutions are not significantly different from the optimum, while in 80% of experiments 4 or more solutions in the top-5 are not significantly different from the optimum. For the same two categories, MO^{2}TOS has only 5% and 29%. These results suggest that the proposed method has a better chance of producing higher quality solutions than MO^{2}OS.

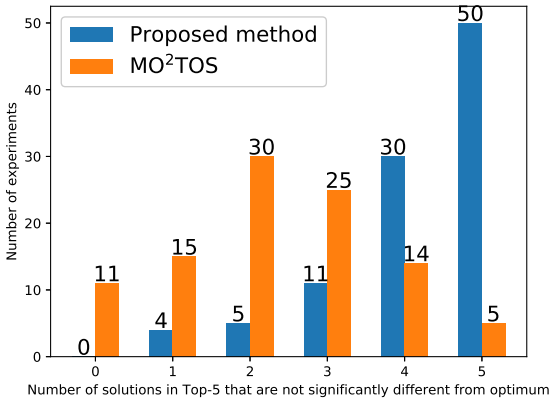

Figure 5: Comparison of the proposed method with MO^{2}TOS over 100 experiments.

5 CONCLUSIONS AND FUTURE WORK

This paper proposes a multi-fidelity simulation optimization algorithm that can be used as part of a symbiotic simulation where the digital twin simulation model is complex and computationally expensive to run. We present preliminary results for a simple inventory model comparing its performance with an exhaustive sampling strategy and MO^2TOS, a leading multi-fidelity algorithm. Results suggest that the proposed method has a higher probability of generating optimization outputs that are not significantly different from the optimal solution.

Future work on the algorithm will consider the trade-off between cluster size and group size before considering adaptive sampling strategies in which the group size of the optimal group reduces as the sampling continues, enabling the algorithm to close in on the optimal set of solutions. We will also carry out further experimentation into the choice of 31 clusters and 5 groups to determine whether better results can be obtained with different values. Currently we use the more traditional OCBA algorithm in which the objective is to maximize the probability of correct selection (PCS) whereas we may see greater benefits in optimizing the Expected Opportunity Cost (EOC) and this is something else to consider.

The optimization algorithm described above fits within a wider framework for symbiotic simulation working with a more complex simulation model and this yields several strands of future work. First, determining how well the method works and how fast it can be on a more complex simulation model. Second, we need to consider how updates to the system parameters and the state of the system will affect the accuracy of the metamodel and the knock-on effect this will have on the ability of the optimization procedure to identify good results.

ACKNOWLEDGEMENTS

This work has been partially funded by Ford Motor Company.

REFERENCES

Aydt, H., S. J. Turner, W. Cai, and M. Y. H. Low. 2008. "Symbiotic Simulation Systems: An Extended Definition Motivated by Symbiosis in Biology". In *Proceedings of the 22nd Workshop on Principles of Advanced and Distributed Simulation*, 109–116. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc,.

Chen, C.-H., and L. H. Lee. 2010. *Stochastic Simulation Optimization: An Optimal Computing Budget Allocation*. Singapore: World Scientific Publishing Company.

Chollet, F. 2015. "Keras". https://keras.io/.

Day, W. H., and H. Edelsbrunner. 1984. "Efficient algorithms for agglomerative hierarchical clustering methods". *Journal of Classification* 1(1):7–24.

Fu, M. C. (Ed.) 2014. *Handbook of Simulation Optimization*. New York: Springer.

Glaessgen, E. H., and D. S. Stargel. 2012. "The digital twin paradigm for future NASA and U.S. Air force vehicles". In *Structures, Structural Dynamics and Materials Conference*, 1–14. Reston, Virginia: American Institute of Aeronautics and Astronautics.

Han, Z. H., and S. Görtz. 2012. "Hierarchical kriging model for variable-fidelity surrogate modeling". *AIAA Journal* 50(9):1885–1896.

Hendrycks, D., and K. Gimpel. 2016. "Gaussian Error Linear Units (GELUs)". *CoRR*:1–9.

Higgins, M., and J. Ladbrook. 2018. "Ford's Power Train Operations: Chaning the Simulation Environment 2". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, 3308–3318. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Horng, S. C., and S. Y. Lin. 2009. "Ordinal optimization of G/G/1/K polling systems with k-limited service discipline". *Journal of Optimization Theory and Applications* 140(2):213–231.

Horng, S. C., and S. Y. Lin. 2013. "Evolutionary algorithm assisted by surrogate model in the framework of ordinal optimization and optimal computing budget allocation". *Information Sciences* 233:214–229.

Huang, D., T. T. Allen, W. I. Notz, and R. A. Miller. 2006. "Sequential kriging optimization using multiple-fidelity evaluations". *Structural and Multidisciplinary Optimization* 32(5):369–382.

Huang, D., T. T. Allen, W. I. Notz, and N. Zeng. 2006. "Global optimization of stochastic black-box systems via sequential kriging meta-models". *Journal of Global Optimization* 34(3):441–466.

Ivanov, D., A. Dolgui, A. Das, and B. Sokolov. 2019. "Digital Supply Chain Twins: Managing the Ripple effect, resilience and disruption risks by data-driven optimization, simulation, and visibility". In *Handbook of Ripple Effects in the Supply Chain*, Volume 276, 309–332. Switzerland: Springer Nature.

Kim, S.-H., and B. L. Nelson. 2006. "On the Asymptotic Validity of Fully Sequential Selection Procedures for Steady-State Simulation". *Operations Research* 54:475–488.

Kingma, D. P., and J. L. Ba. 2015. "Adam: A method for stochastic optimization". *CoRR*:1–15.

Law, A. M. 2015. *Simulation Modeling and Analysis. Fifth Edition*. 5th ed. New York: McGraw-Hill Education.

Li, H., Y. Li, L. H. Lee, E. P. Chew, G. Pedrielli, and C. H. Chen. 2016. "Multi-objective multi-fidelity optimization with ordinal transformation and optimal sampling". In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. M. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, 3737–3748. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Lin, Z., A. Matta, and J. G. Shanthikumar. 2019. "Combining simulation experiments and analytical models with area-based accuracy for performance evaluation of manufacturing systems". *IISE Transactions* 51(3):266–283.

Moore, R. A., D. A. Romero, and C. J. Paredis. 2014. "Value-based global optimization". *Journal of Mechanical Design, Transactions of the ASME* 136(4):1–14.

Morris, M. D., and T. J. Mitchell. 1995, feb. "Exploratory designs for computational experiments". *Journal of Statistical Planning and Inference* 43(3):381–402.

Negria, E., L. Fumagallia, and M. Macchi. 2017. "A review of the roles of Digital Twin in CPS-based production systems". In *27th International Conference on Flexible Automation and Intelligent Manufacturing*, edited by M. Pellicciari and M. Peruzzini, Number 11, 939–948. Modena, Italy: Elsevier B.V.

Ojha, V. K., A. Abraham, and V. Snášel. 2017. "Metaheuristic design of feedforward neural networks: A review of two decades of research". *Engineering Applications of Artificial Intelligence* 60:97–116.

Onggo, B. S. 2019. "Symbiotic Simulation System (S3) for Industry 4.0". In *Simulation for Industry 4.0: Past, Present and Future*, edited by M. Gunal, 153–165. Switzerland: Springer International Publishing.

Onggo, B. S., C. G. Corlu, J. A. Angel, T. Monks, and R. de la Torre. 2021. "Combining symbiotic simulation systems with enterprise data storage systems for real-time decision-making". *Enterprise Information Systems* 15(2):230–247.

Osorio, C., and M. Bierlaire. 2013. "A simulation-based optimization framework for urban transportation problems". *Operations Research* 61(6):1333–1345.

Pedrielli, G., K. Selcuk Candan, X. Chen, L. Mathesen, A. Inanalouganji, J. Xu, C. H. Chen, and L. H. Lee. 2019. "Generalized Ordinal Learning Framework (GOLF) for Decision Making with Future Simulated Data". *Asia-Pacific Journal of Operational Research* 36(6):1–35.

Rhodes-Leader, L., D. J. Worthington, B. L. Nelson, and B. S. Onggo. 2018. "Multi-fidelity simulation optimisation for airline disruption management". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, 2179–2190. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Song, J., Y. Qiu, J. Xu, and F. Yang. 2019. "Multi-fidelity sampling for efficient simulation-based decision making in manufacturing management". *IISE Transactions* 51(7):792–805.

Xu, J., E. Huang, L. Hsieh, L. H. Lee, Q. S. Jia, and C. H. Chen. 2016. "Simulation optimization in the era of Industrial 4.0 and the Industrial Internet". *Journal of Simulation* 10(4):310–320.

Xu, J., S. Zhang, E. Huang, C.-H. Chen, L. H. Lee, and N. Celik. 2014, dec. "Efficient multi-fidelity simulation optimization". In *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. J. Buckley, and J. A. Miller, 3940–3951. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Zhang, X., and W. Zhu. 2019. "Application framework of digital twin-driven product smart manufacturing system: A case study of aeroengine blade manufacturing". *International Journal of Advanced Robotic Systems* 16(5):1–16.

AUTHOR BIOGRAPHIES

YIYUN CAO is a PhD student at the University of Southampton and a member of the Centre for Operational Research, Management Sciences and Information Systems (CORMSIS). His research interests include simulation optimization, data analysis and machine learning. His email address is yiyun.cao@soton.ac.uk.

CHRISTINE CURRIE is a Professor of Operational Research in Mathematical Sciences at the University of Southampton and a member of the Centre for Operational Research, Management Sciences and Information Systems (CORMSIS). She is Editor-in-Chief for the Journal of Simulation. Her email address is christine.currie@soton.ac.uk and her website is http://www.southampton.ac.uk/maths/about/staff/ccurrie.page.

BHAKTI STEPHAN ONGGO is a Professor of Business Analytics at the Centre for Operational Research, Management Science and Information Systems (CORMSIS), Southampton Business School, University of Southampton, UK. He is the Associate Editor for the Journal of Simulation, Area Editor for Health Systems, and chair of The OR Society's Simulation SIG. His email address is b.s.s.onggo@soton.ac.uk and his website is http://bsonggo.wordpress.com.

MICHAEL HIGGINS is the Simulation and Process Optimisation Leader for Ford where he has worked since 2012. In 2014 he gained an MPhil(Eng) through his research into the application of Discrete Event Simulation within the automotive industry. He is currently researching and developing methods to increase usability, and widen the application of, the Ford Simulation Toolset in-line with Industry 4.0. His email address is michael.higgins@ford.com.