Journal of Industrial Engineering International

An artificial neural network meta-model to solve semi-expensive simulation optimization problems: A comparative study

--Manuscript Draft--

Manuscript Number:	
Full Title:	An artificial neural network meta-model to solve semi-expensive simulation optimization problems: A comparative study
Article Type:	Original Research
Funding Information:	
Abstract:	Although a considerable number of problems whose analysis depends on a set of complex mathematical relations exist in the literature due to recent developments in the field of decision making, still very simplified and unrealistic assumptions are involved in many. Simulation is one of the most powerful tools to deal with this kind of problems and enjoys being free of any restricting assumptions which may generally be considered in a stochastic system. In addition, simulation optimization techniques are categorized into two broad classes of model-based and meta-model-based methods. In the first class, simulation and optimization component interact with each other causing an increase in simulation times and costs. To cope with this problem, a third component defined as a meta-model that estimates the relationships between the inputs and outputs of the system being simulated comes to the picture in the second class problems. Besides, optimization of semi-expensive simulation optimization problems needs a numerous simulation run in model-based methods. However, as the validation cost increases at a rapid rate in each iteration of the meta-model-based methods, a new method which consists of two phases has been introduced in the literature to solve semi-expensive simulation optimization problems in less computational time. In the first phase, similar to a model-based algorithm, the simulation output is used directly in the optimization stage. In the second phase, the simulation model is changed with a validated meta-model. In this paper, an artificial neural network is employed as the meta-model in order to compare its performance to the ones of the original algorithm that uses a Kriging meta-model in five popular test problems as well as an (s, S) inventory problem.
Corresponding Author:	S.T.A. Niaki, Ph.D. Sharif University of Technology Tehran, Tehran IRAN, ISLAMIC REPUBLIC OF
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	Sharif University of Technology
Corresponding Author's Secondary Institution:	
First Author:	Mohammad Behbahani, M.Sc.
First Author Secondary Information:	
Order of Authors:	Mohammad Behbahani, M.Sc.
	S.T.A. Niaki, Ph.D.
	Mehran Moazeni, M.Sc.
Order of Authors Secondary Information:	
Author Comments:	No comments. Thanks

An artificial neural network meta-model to solve semi-expensive simulation optimization problems: A comparative study

Mohammad Behbahani, M.Sc.
Department of Industrial Engineering, Sharif University of Technology
Email: Behbahanimd@gmail.com

Seyed Taghi Akhavan Niaki[1], Ph.D.
Department of Industrial Engineering, Sharif University of Technology
P.O. Box 11155-9414, Azadi Ave., Tehran 1458889694 Iran
Email: Niaki@Sharif.edu

Mehran Moazeni, M.Sc.
Department of Industrial Engineering, Sharif University of Technology
Email: moazeni.mehran@gmail.com

Abstract

Although a considerable number of problems whose analysis depends on a set of complex mathematical relations exist in the literature due to recent developments in the field of decision making, still very simplified and unrealistic assumptions are involved in many. Simulation is one of the most powerful tools to deal with this kind of problems and enjoys being free of any restricting assumptions which may generally be considered in a stochastic system. In addition, simulation optimization techniques are categorized into two broad classes of model-based and meta-model-based methods. In the first class, simulation and optimization component interact with each other causing an increase in simulation times and costs. To cope with this problem, a third component defined as a meta-model that estimates the relationships between the inputs and outputs of the system being simulated comes to the picture in the second class problems. Besides, optimization of semi-expensive simulation optimization problems needs a numerous simulation run in model-based methods. However, as the validation cost increases at a rapid rate in each iteration of the meta-model-based methods, a new method which consists of two phases has been introduced in the literature to solve semi-expensive simulation optimization problems in less computational time. In the first phase, similar to a model-based algorithm, the simulation output is used directly in the optimization stage. In the second phase, the simulation model is changed with a validated meta-model. In this paper, an artificial neural network is employed as the meta-model in order to compare its performance to the ones of the original algorithm that uses a Kriging meta-model in five popular test problems as well as an (*s*, *S*) inventory problem.

Keywords: Semi-expensive simulation problems; Simulation optimization; Meta-model-based algorithm; Artificial neural network

[1] Corresponding author

An artificial neural network meta-model to solve semi-expensive simulation optimization problems: A comparative study

Abstract

Although a considerable number of problems whose analysis depends on a set of complex mathematical relations exist in the literature due to recent developments in the field of decision making, still very simplified and unrealistic assumptions are involved in many. Simulation is one of the most powerful tools to deal with this kind of problems and enjoys being free of any restricting assumptions which may generally be considered in a stochastic system. In addition, simulation optimization techniques are categorized into two broad classes of model-based and meta-model-based methods. In the first class, simulation and optimization component interact with each other causing an increase in simulation times and costs. To cope with this problem, a third component defined as a meta-model that estimates the relationships between the inputs and outputs of the system being simulated comes to the picture in the second class problems. Besides, optimization of semi-expensive simulation optimization problems needs a numerous simulation run in model-based methods. However, as the validation cost increases at a rapid rate in each iteration of the meta-model-based methods, a new method which consists of two phases has been introduced in the literature to solve semi-expensive simulation optimization problems in less computational time. In the first phase, similar to a model-based algorithm, the simulation output is used directly in the optimization stage. In the second phase, the simulation model is changed with a validated meta-model. In this paper, an artificial neural network is employed as the meta-model in order to compare its performance to the ones of the original algorithm that uses a Kriging meta-model in five popular test problems as well as an (*s*, *S*) inventory problem.

Keywords: Semi-expensive simulation problems; Simulation optimization; Meta-model-based algorithm; Artificial neural network

1. Introduction

From its very beginnings over five decades ago, simulation has been a powerful tool to assess potential risks and to guide managers and practitioners in making decisions under uncertainty. The simulation approach can provide more accurate anticipating risks and makes more robust decisions in the face of uncertainty, ambiguity, and variability. Moreover, the process of finding the best values of input variables among all possibilities without explicitly evaluating each possibility is the so-called simulation optimization. There is a wide variety of the applicability of the simulation-optimization approach in different problems. For instance, a few applications of the simulation-optimization approach include liver transplant management, maritime logistics optimization, semiconductor production planning, and parasitology calibration mentioned in Xu et al. (2015).

There are basically two main methods to optimize a simulation model; (1) model-based and (2) meta-model-based. In the first method, simulation and optimization phases are performed in an iterative approach until a stopping criterion is met. In the latter, however, a surrogate model is used to mimic the simulation behavior when a high computational burden such as the one in the Ford Motor Company crash simulation, which takes about 36-60 hours for one replication, is involved (Wang & Shan, 2007). Having one crash in each iteration of this simulation and assuming that an average number of 50 iterations are needed for the optimization of a two-variable problem, the total computational time would range between 75 days to 11 months, which is unacceptable in practice. Nevertheless, predicting an approximate meta-model (surrogate model) that replaces the simulation model would result in a negligible cost compared to the above simulation cost (Wang et al., 2019). The widely used surrogate models in the literature include Kriging, Artificial Neural Networks (ANN), Support Vector Machines (SVM), and Radial Basis Function (RBF).

Establishing a surrogate model often goes through three steps as follows (Nguyen et al., 2014):

- Sampling input vectors and calculating corresponding model responses, which constitute a database for training a surrogate model.
- Constructing the surrogate model based on the database by selecting an appropriate method, e.g., Kriging, SVM, ANN, RBF.
- Validating the model before being used as a ‘‘surrogate’’ of the original model.

Both model and meta-model based methods have some benefits and drawbacks summarized in Table 1. Although meta-model based algorithms have some disadvantages, when a single replication of the simulation model takes more than about five minutes, one has to use a meta-model based algorithm. This is due to the fact that a large number (usually 2000 to 4000 on the average) of replications is required in model-based simulations to assess simulation points in order to use them is the optimization process. In a meta-model-based simulation, however, the number of simulation replications is minimized. Meanwhile, when a single replication of the simulation model takes less than five minutes, meta-model approaches are not efficient as they require too much time for their validation processes and due to their fitting errors. In these cases, the use of the model-based approach is recommended.

Table 1: Pros and Cons of simulation optimization approaches

Method	Advantages	Disadvantages
Model-Based	• Accurate and numerically efficient for inexpensive simulation problems	• Needs to assess a large number of simulation points • Gives no insight about the objective function
Meta-Model Based	• Relieve the computational expense by replacing the simulation with an approximation model	• Existence of fitness error • Validation time • Gets trapped in the fitting and validating steps

Sometimes a single replication of a simulation model takes more than two, but less than 5 minutes on the average. As such, neither the model-based nor the meta-model-based simulation methods are appropriate due to their disadvantages. For this kind of simulations, Moghaddam & Mahlooji (2017) presented an algorithm named Semi-Metamodel-Based (SMB) that employs a meta-model while it is different from the common meta-model-based algorithms. They showed that their algorithm does not have the disadvantages of the model-based approach and tends to alleviate some of the problems of the meta-model-based algorithms. In the first phase, the algorithm acts as a model-based method and directly uses the simulation output in the optimization phase. In this phase, the optimization component goes forward and the construction of the meta-model gets underway in parallel. As in every step, several simulation points are added to the experimental design, the time required for the meta-model validation decreases. The second phase of the algorithm employs the meta-model obtained from the first phase to evaluate the solutions in the optimization stage and begins once the meta-model is validated. Furthermore, they presented an optimization algorithm based on particle swarm optimization (PSO) which employs some strategies to improve its intensification and diversification characteristics.

The primary approach used to construct a surrogate model based on available simulation points is the Gaussian process regression or the so-called Kriging method. This estimation method aims to find a minimum error-variance estimate of any un-sampled simulation point by smoothing out the extreme values of the available simulation points. As such, the objective of the current paper is to compare the performances of the Kriging and an ANN approach used as surrogate models in the SMB algorithm (Moghaddam & Mahlooji, 2017). To this aim, simulation points obtained by an (s, S) inventory control model as well as the ones from five popular test functions are taken into the investigation.

The structure of the rest of the paper is as follows. The literature review is given in details in Section 2. The employed ANN meta-model is described in Section 3. Section 4 contains the characteristics of the SMB algorithm. The (s, S) inventory and the test functions are brought in Section 5 to assess the performance of ANN meta-model in the SMB algorithm and to demonstrate the comparison of its results with the ones obtained by the Kriging-based meta-model. Finally, the paper is concluded in Section 6 where some recommendations are given for future research.

2. Literature review

In this section, some of the available model-based and meta-model-based algorithms used to solve simulation optimization problems are first surveyed. Then, the basic differences between the Kriging and the ANN meta-model in terms of the simulation-optimization techniques will be discussed. As mentioned above, several model-based approaches have been proposed in the literature on simulation optimization problems. Wang (2005) worked an excellent optimization approach using a hybrid method of genetic algorithm (GA) and neural network. They employed a neural network to predict the objective function value and then utilized a GA to use its effective and robust evolutionary searching ability in determining the best values of the input parameters. Shi & Ólafsson (2000) proposed the nested partition as a global sampling method that continuously is adapted via partitioning of the feasible solution region, attempting to reduce the computational load by selecting the most promising regions. Besides, Ahmed & Alkhamis (2009) integrated simulation with optimization to design a decision support tool for a governmental hospital in Kuwait. They evaluated the impacts of various staffing levels on hospital service efficiency in order to determine the optimal number of doctors. Tsai & Fu (2014) employed two GA-based algorithms for a discrete simulation optimization problem with a single stochastic constraint that adopts different sampling rules and searching mechanisms, and thus deliver various statistical guarantees.

Meta-modeling has been applied to build simulations for a variety of reasons including early decision-making designs, uncertainty and sensitivity analyses, design optimizations, and model calibrations (Nguyen et al., 2014). The meta-model-based technique is one of the most popular research areas in the simulation optimization field, where various algorithms have been proposed. For instance, Jones et al. (1998) used the Kriging approach in an algorithm called Efficient Global Optimization (EGO) to interpolate between function values and chose future samples based on an expected improvement metric for simulations with a deterministic output. In addition, ANNs have been employed in many algorithms proposed in the literature to optimize time-consuming simulations. To name a few, Dengiz et al. (2009) optimized two manufacturing systems utilizing neural network meta-models in which a tabu search (TS) meta-heuristic was used for the training of the ANNs in order to improve the performance of the meta-modeling approach. Mohammad Nezhad & Mahlooji (2014) presented an ANN meta-model for expensive continuous simulation optimization (SO) with stochastic constraints. Capturing the non-linear nature of the ANN, the SO problem was iteratively approximated via non-linear programming problems whose (near) optimal solutions obtain estimates of the global optima. A comprehensive review of other meta-models for optimization strategies as computationally-expensive black-box functions is available in Shan & Wang (2010).

Neural network and Kriging approximation are among the most attractive techniques in simulation-optimization meta-modeling. Willmes et al. (2003) compared the performance of feed-forward neural networks and Kriging as fitness approximation used in evolutionary optimization in off-line and online learning. Yuan & Guangchen (2009) applied four performance measures to evaluate different types of meta-model performances such as the ability to provide good starting points for gradient-based search, the accuracy of placing optima in the correct location and so on. Furthermore, the performance of the Kriging model was compared with the one of an ANN meta-model in Vicario et al. (2016) in order to determine which model guarantees a higher accuracy in predicting the result of four-dimensional computational fluid dynamics system. Moreover, Beşikçi et al. (2016) compared the performance of an ANN meta-model with one of a multiple regression (MR) model, when the superiority of the former was confirmed. Interested readers are referred to Østergård et al. (2018) for other comparison studies among linear regression with ordinary least squares (OLS), random forest (RF), support vector regression (SVR), multivariate adaptive regression splines, Gaussian process regression (GPR), and neural network meta-model. In their work, the authors considered five performance indicators to be used in eight test problems as well as 19 mathematical test functions.

3. Neural network meta-model

Neural networks (NNs) are powerful tools for the approximation of unknown nonlinear functions and have gained wide applications in a variety of fields. ANNs can approximate arbitrary smooth functions and can be fitted using noisy response values. ANNs were developed to mimic neural processing and can be implemented on a digital computer using networks of numerical processors whose inputs and outputs are linked according to specific topologies (Barton & Meckesheimer, 2006).

NNs used for function approximation are typically multi-layer feed-forward networks. A feed-forward ANN refers to an ANN architecture in which signals flow towards the output layer in a forward manner. Feed-forward layered networks have the flexibility to approximate smooth functions arbitrarily well, provided sufficient nodes and layers. Multilayer ANNs are usually capable of modeling more difficult problems.

Let the following tan-sigmoid ($f_{tan}(x)$) and linear ($f_{lin}(x)$) functions be the transfer functions for the hidden and output layers, respectively,

$$f_{tan}(x) = \frac{2}{1+exp(-2x)} - 1 \tag{1}$$

$$f_{lin}(x) = x. \tag{2}$$

Then, Figure 1 represents a sample of two layered feed-forward ANN. In this network, each neuron in a layer is linked only with neurons of a different layer and the outputs are determined by the following two equations:

$$z_s^k(x) = \frac{2}{1+exp\left(-2\left(\sum_{j=1}^{d} v_{js}^0 x_j + b_s^k\right)\right)} - 1 \qquad ; \qquad k = 1, s = 1, \dots, nn_k \tag{3}$$

$$y(x) = \sum_{s=1}^{nn_{k-1}} v_{s's}^{k-1} z_{s'}^{k-1}(x) + b_s^k \qquad ; \qquad k = 1, s = 1, \tag{4}$$

where x_j serves as input neuron j, v_{js}^0 denotes the weight of the connection link between input neuron j and hidden neuron s in the first layer, and $v_{s's}^{k-1}$ denotes the weight of the connection link between neuron s' in the hidden layer $k-1$ and neuron s in the layer k. In addition, b_1^2 denotes the bias value of the output neuron and $z_s(x)$ is the activation value of the hidden neuron s. A training method defined on a feed-forward ANN to solve such mapping problems modifies the weights and biases in such a manner that the following performance criterion is minimized:

$$MSE = \frac{\sum_{i=1}^{n} \left(\overline{w}(x_i) - y(x_i)\right)^2}{n}, \tag{5}$$

where x_i for $i = 1, \dots, n$ stands for the input patterns, $\overline{w}(x_i)$ denotes the target outputs and n shows the number of input patterns (Mohammad Nezhad & Mahlooji, 2014).

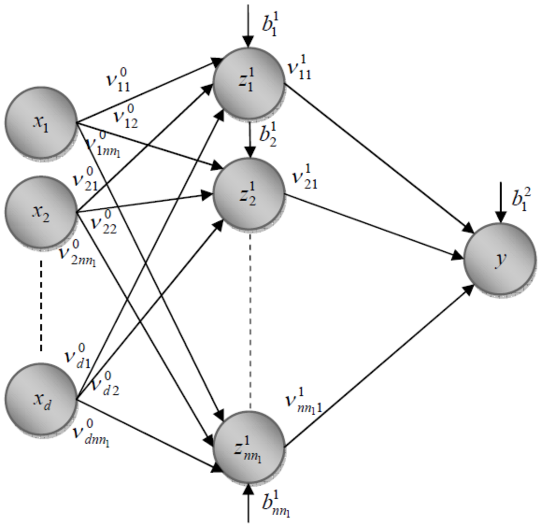

Figure 1. A sample of a two-layered feed-forward artificial neural network

3.1. ANN versus Kriging

As stated in Section 2, there are some works in the literature that compare ANN with Kriging and other simulation meta-models (see for example Wang & Shan, 2007; Willmes et al., 2003; Yuan & Guangchen, 2009; Vicario et al., 2016). The comparisons are performed with respect to one or some of the efficiency, accuracy, interpretability and robustness performance measures. Some of the advantages and disadvantages of Kriging and ANN meta-models are reported in Table 2 (Østergård et al., 2018).

Table 2: Comparison of Kriging and ANN meta-models

Meta-model	Advantages	Disadvantages
Kriging	• less sensitive to the chosen settings	• among the slowest algorithms • becomes unstable for large training sets
ANN	• For the most accurate, non-linear methods, NN proved the most efficient for large training sets • less time-consuming for new predictions	• have a large number of possible configurations • the least interpretable method • accuracies obtained for NN vary when repeating the meta-modeling

As seen in Table 2, determining the best combination of the neurons in the hidden layer, the weights, and the biases, referred to a configuration, as well as choosing a transfer function and the training algorithm is not simple in an ANN-based meta-model. Although this makes it difficult to conduct a fair comparison between ANN and other meta-models, there are some heuristic methods to determine the ANN hyper-parameters. For example, Rigoni & Lovison (2007) stated that for a network with the number of training data q time the number of output variables m, the hidden layer neurons h could be calculated by:

$$h \leq fix\left(\frac{m(q-1)}{(n+m+1)}\right), \quad (6)$$

where $fix(.)$ is the greatest integer less than or equal to its argument and n denotes the number of input variables.

4. The proposed ANN semi-meta-model-based algorithm

A semi-meta-model-based algorithm was introduced for the first time by Moghaddam & Mahlooji (2017) to deal with the model-based and meta-model-based difficulties in semi-expensive simulation optimization problems which take about 2-5 minutes. The main meta-model used in their work to estimate the relationship between the input and the output of the simulation model was the Kriging. Furthermore, they utilized a new particle swarm optimization

(PSO) algorithm with improved exploration and exploitation characteristics. In what comes next, while the use of an ANN-based meta-model, instead of Kriging, is justified, a summary of the steps involved in their research and the differences between their work and the current paper is highlighted.

Simulation outputs have been used not only in meta-model fitting and validation steps but also in the optimization stage. This means that the meta-model fitting and optimization components do not longer work independently. Besides, if the validity of a meta-model is rejected, instead of one new point a few simulation points are added in order to fit and validate expensive simulation problems. As such, the processing time could be shortened significantly when an ANN-based meta-model is used to solve semi-expensive simulation optimization problems.

The flowchart of the proposed ANN semi-meta-model-based algorithm that is shown in Figure 2 consists of two phases. In the first phase, the algorithm enjoys model-based characteristics where the output of the simulation model is directly used for optimization purposes. In the second phase, the algorithm acts as a meta-model-based algorithm where the validated meta-model obtained from the first phase will substitute for the simulation model. Furthermore, similar to Moghaddam & Mahlooji (2017), a spatial hole-PSO (SH-PSO) is introduced and used in both phases to attract newly generated particles towards empty areas of the solution space.

4.1. Constructing the ANN meta-model

As the Levenberg-Marquardt optimization is one of the fastest back-propagation algorithms, it is used in this paper as the network training function. This algorithm has been designed to approach second-order training speed without having to compute the Hessian matrix. Although it does require more memory than the other training algorithms, it is especially suited for function approximation problems where the network contains up to several hundreds weights and the approximation need to be very accurate (Yuan & Guangchen, 2009). In addition, 70% of the data at hand are assigned to the training process and the remaining to the validation test according to MATLAB user’s recommendations.

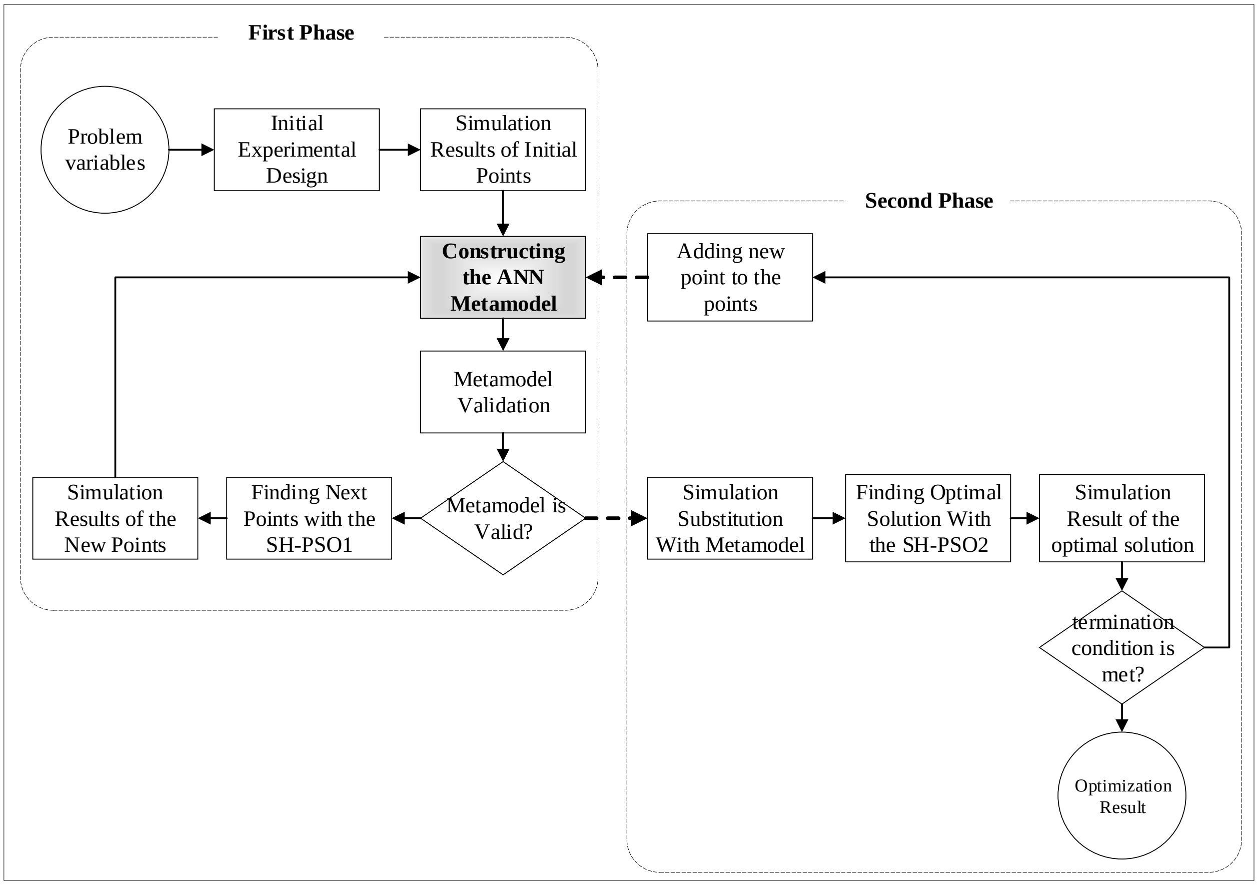
First Phase
Problem variables
Initial Experimental Design
Simulation Results of Initial Points
Constructing the ANN Metamodel
Metamodel Validation
Metamodel is Valid?
Finding Next Points with the SH-PSO1
Simulation Results of the New Points
Second Phase
Adding new point to the points
Simulation Substitution With Metamodel
Finding Optimal Solution With the SH-PSO2
Simulation Result of the optimal solution
termination condition is met?
Optimization Result

Figure 2. The flowchart of the ANN semi-meta-model-based algorithm

4.2. Algorithm specifications

The comparison of the proposed ANN-based and the Kriging (Moghaddam & Mahlooji, 2017) meta-models should be fair. Thus, most of the specifications of an ANN-based semi-meta-model-based algorithm such as initial experimental design, the employed PSO algorithm, the meta-model validation process, and so on must be as much as similar to the ones used in Moghaddam & Mahlooji (2017). In what follows these specifications are presented.

For the initial sampling points, the Latin hypercube sampling method that maximizes the minimum distances between the points is used in both methods. Moreover, the leave-one-out cross-validation approach is used as the validation scheme in both algorithms. In this approach, the value of the studentized prediction error for every output of the left-out point *i* is calculated as

$$t_{r-1}^{i} = \frac{\bar{\bar{w}}(x_i) - \bar{y}^*(x_i)}{\sqrt{\widehat{var}(\bar{\bar{w}}(x_i)) + \widehat{var}(y^*(x_i))}}, \qquad (7)$$

where $\overline{\overline{w}}(x_i)$ is the average of the bootstrapped simulation outputs and $\bar{y}^*(x_i)$ represents the average of the bootstrapped ANN predictors of the left-out point x_i. Furthermore, $\widehat{var}(\overline{\overline{w}}(x_i))$ and $\widehat{var}(y^*(x_i))$ are the variance of the averages of the bootstrapped simulation outputs and the variance of the bootstrapped ANN predictors of x_i, respectively.

The main differences between the SH-PSO algorithms with the simple PSO are the SH-detecting algorithm used to find the best points and to attract newly generated particles towards themselves with the help of assigning a weight between zero and one (regarding the objective function) to each spatial hole. Besides, the SH-PSO algorithm uses a simulation model outside the algorithm for solution evaluation in the first phase. Contrariwise, in the second phase, the ANN meta-model works inside the algorithm and is used for the evaluation process. Interested readers are referred to Moghaddam & Mahlooji (2017) for more information on the SH-detecting and the SH-PSO algorithms.

5. Applications and results

An (s, S) inventory control model alongside five popular test functions is used as simulation models in this section in order to assess the performances of the two semi-meta-model-based algorithms discussed in Section 4.

5.1. The (s, S) inventory as a realistic simulation model

An (s, S) inventory control model is provided in this section to compare the performances of the proposed ANN and the existing Kriging as the simulation meta-models in the SMB algorithm. In this model, a replenishment order is placed as soon as the inventory position (on-hand inventory + outstanding orders – backlogs) drops to or goes below the reorder point *s*. This replenishment order brings the inventory position back to the order-up-to level *S*. Note that *S* consequently denotes the maximum inventory position. The parameters of this (*s*, *S*) inventory control model are as follows (Biles at al., 2007):

- The holding cost is charged $1 per day per item
- The shortage cost is charged $5 per day per item
- The simulation period is 4,000 days per replicate
- The ordering cost is $32 plus $3 per unit ordered
- The order arrival time follows an exponential distribution with a mean of 6 days

- The inventory position is reviewed at the end of each day
- The customer demand is exponentially distributed with the mean of 90
- The number of units demanded per customer is 1.

The value of the parameters of the optimization processes ANN-SMB and Kriging-SMB are set according to Table 3. Both of the algorithms are coded in MATLAB 8.4 environment. Besides, in order to have a comfortable linkage between the proposed simulation and the optimization models, the inventory model is simulated in MATLAB 8.4 as well.

Table 3: The parameters of the ANN-SMB and the Kriging-SMB algorithms to solve the (s, S) inventory control problem.

Parameter	Value	
	ANN-SMB Algorithm	Kriging-SMB Algorithm
Number of initial design points (idp)	20	10
Number of simulation replication (r)	4	4
Particle size (m)	10	10
Maximum number of spatial holes (sh^{max})	5	5
Maximum number of neighbors to be evaluated (N^{max})	15	10
Maximum acceptable error for meta-model validation	0.05	0.1
Number of bootstrap replications for meta-model validation	100	100
The strength of the movement towards the local best (c_1)	2	2
The strength of the movement towards the global best (c_2)	2	2
Terminating condition (T_{min})	0.005	0.005

The results of 10 independent replications of the simulation-optimization are shown in Table 4 for both the ANN and the Kriging semi-meta-model-based algorithms. The obtained solutions and the objective functions with the number of solution evaluations are presented in this table.

As seen in Table 4, the average of the optimal values of the ANN-SMB algorithm is 595.28 with a standard deviation of 10.61. These values are 603.71 and 14.99 for the Kriging-SMB algorithm, respectively. Here, the null hypothesis $\mu_{ANN-SMB} \leq \mu_{Kriging-SMB}$ is tested using a two-sample student t-test in order to determine if the obtained average optimal value of the ANN-SMB is not greater than that of the kriging-SMB significantly. As the t-statistics and the p-value are obtained as -1.4782 and 0.9133, respectively, the null hypothesis cannot be rejected at a 5% significance level. This implies that the two algorithms provide equal quality solutions on the average. Based on the results shown in the fourth and sevenths columns of Table 4, a similar test of hypothesis is performed here to compare the average number of simulation evaluations involved in the above two algorithms. As the P-value of the two-sample t-test is obtained as 0.953, again the null hypothesis on the equality of the two means cannot be rejected at the 5% significance level. This implies that the average number of simulation evaluations used in the two algorithms does not differ statistically.

Table 4: Comparison of ANN-SMB and Kriging-SMB algorithms for the (s, S) inventory control problem.

Methods	ANN-SMB Algorithm			Kriging-SMB Algorithm		
Run	Solution (s, S)	Objective	No. of Simulation Evaluations	Solution (s, S)	Objective	No. of Simulation Evaluations
1	(741, 845)	604.58	195	(678, 855)	606.97	280
2	(658, 894)	611.75	285	(702, 880)	599.64	260
3	(712, 870)	590.07	240	(692, 813)	613.95	280
4	**(721, 876)**	**584.29**	330	(696, 848)	594.43	240
5	(682, 824)	605.16	285	(628, 796)	635.41	280
6	(736, 873)	587.54	285	(717, 886)	595.96	280
7	(665, 835)	601.58	150	**(749, 895)**	**584.48**	240
8	(740, 887)	578.24	375	(662, 857)	617.38	240
9	(696, 857)	591.68	330	(741, 861)	591.97	260
10	(709, 874)	597.92	150	(730, 848)	596.91	280
Avg.		595.28	262.5		603.71	264
Std.		10.61			14.99	

The best (s, S) and the best objective values are shown in bold.

5.2. Analytical test functions

In this section, the performances of the proposed ANN-SMB and the existing Kriging-SMB algorithms are compared to each other using five popular single-objective optimization test problems called Sphere, Griewank, Schaffer's F6, Rastrigin, and Rosenbrock (Molga & Smutnicki, 2005). The solutions obtained using the ANN-SMB algorithm on these test functions alongside its comparative analysis with the Kriging-SMB solutions obtained in Moghaddam & Mahlooji (2017) are presented in Table 5. In this table, the values of $idp,\ r,\ m,$ and sh^{max}

parameters are respectively determined 20, 5, 10, and 5 for the ANN-SMB algorithm and 10, 5, 10, and 5 for the Kriging-SMB algorithm. The other parameter values are the same as the ones used in Table 3. In addition, while the optimal solutions of the functions are shown in the third column, the fourth and fifth column contains respectively the sample means and the sample standard deviations (Std.) of the solutions obtained by the proposed ANN-SMB algorithm. Similarly, the sample means and the sample standard deviations of the solutions obtained by the Kriginig-SMB algorithm are shown in the seventh and the eights column of Table 5, respectively. Finally, the required numbers of function evaluations of the ANN-SMB and the Kriging-SMB algorithms to reach the solutions are reported in the sixth and ninth columns, respectively.

As seen in Table 5, while the proposed ANN-SMB algorithm works better than the Kriging-SMB algorithm for the Sphere, the Griewank, the Schaffer's F6, and the Rosenbrock test functions on the average, the optimization results obtained for the Rastrigin function shows that the Kriging-SMB performs better in this case. Moreover, the result of a two-sample student t-test for the equality of the average numbers of function evaluations involved in the two algorithms shows that there are no significant differences between the two algorithms (P-value=0.570). This implies that the proposed algorithm is the better one in general. Besides, the outcome of a one-way blocked (the 5 test problems are used as blocks) analysis of variance (ANOVA), which is used to test the equality of the performance means of the two algorithms in all test problems, is reported in Table 6 based on 20 replications in each block. Once again, as the null hypothesis is not rejected at a *p*-value of 0.321, the two algorithms do not differ significantly.

6. Conclusion and future works

This paper presented a new version of a semi-meta-model-based simulation optimization algorithm, in which an artificial neural network was used as the primary meta-model. Although the results obtained using statistical comparisons with a Kriging-based meta-model did not show a significant difference between the performances of the proposed ANN-SMB and the Kriging-SMB algorithms for five popular test functions, we showed that the ANN-SMB works as the better algorithm to optimize four out of five test functions as well as for an (s, S) inventory optimization model.

Table 5: *Nmerical results of using the ANN-SMB and the Kriging-SMB algorithms on five test problems*

			ANN-SMB Algorithm			Kriging-SMB Algorithm		
			Objective			Objective		
Function	Formula	Optimal	Mean	Std.	Number of function evaluations	Mean	Std.	Number of function evaluations
Sphere	$y = \sum_{j=1}^{d} x_j^2$ $d = 2, -2 \le x_j \le 2, j = 1,2$	$x^* = [0,0]$ $y^* = 0$	0.018	0.024	266	0.189	0.09	244
Griewank	$y = \frac{1}{4000}\sum_{j=1}^{d} x_j^2 - \prod_{j=1}^{d} \cos\left(\frac{x_j}{\sqrt{j}}\right) + 1$ $d = 2, -8 \le x_j \le 8, j = 1,2$	$x^* = [0,0]$ $y^* = 0$	0.024	0.017	415	0.335	0.22	368
Schaffer's F6	$y = 0.5 - \frac{\left(\sin\sqrt{(x_1^2 + x_2^2)}\right)^2 - 0.5}{\left(1 + 0.001(x_1^2 + x_2^2)\right)^2}$ $d = 2, -3 \le x_j \le 3, j = 1,2$	$x^* = [0,0,0]$ $y^* = 0$	0	0	492	0.501	0.05	425
Rastrigin	$y = \sum_{j=1}^{d}\left(x_i^2 - 10\cos(2 \times \pi x_j) + 10\right)$ d= $3, -5 \le x_j \le 5,\ j = 1,2,3$	$x^* = [0,0,0]$ $y^* = 0$	4.317	1.122	460	0.772	0.16	457
Rosenbrock	$y = \sum_{j=1}^{d-1}\left(100(x_{j+1} - x_j^2)\right)^2 + (x_j - 1)^2$ $d = 3, -3 \le x_j \le 3, j = 1,2,3$	$x^* = [1,1,1]$ $y^* = 0$	2.133	1.142	528	3.103	1.67	483
Avg.					432.2			395.4

Table 6: Analysis of variance to compare the performances of the ANN-SMB and the Kriging-SMB algorithms on the five test functions

Source	DF	Adj. SS	Adj. MS	F-Value	P-Value
Treatment	1	59,577	59,577	1.00	0.321
Block	4	247,438	61,860		
Error	94	5,626,298	59,854		
Total	99	5,933,313			

As the number of parameters and the tradeoff between the accuracy and the required speed is a critical factor to decide which algorithm to use, it is highly recommended to take into account the benefits and the drawbacks of each meta-model.

While the present study did not examine the performances of some other popular meta-models such as radial basis function (RBF), regression models, and so on in the SMB algorithm, their use along with their comparisons with the ones investigated in this paper is recommended for a future investigation. Furthermore, changing or modifying the meta-heuristic algorithm in the optimization steps could have a positive effect on the performances of various meta-models. Comparing the performances of different versions of SMB with real-world semi-expensive problems is another suggestion.

References

Ahmed, M. A., Alkhamis, T. M. (2009). "Simulation optimization for an emergency department healthcare unit in Kuwait," *European Journal of Operational Research*, 198 (3): 936-942.

Barton, R. R., Meckesheimer, M. (2006). "Metamodel-based simulation optimization," *Handbooks in Operations Research and Management Science*, 13: 535-574.

Beşikçi, E. B., Arslan, O., Turan, O., Ölçer, A. (2016). "An artificial neural network based decision support system for energy efficient ship operations," *Computers & Operations Research*, 66: 393-401.

Biles, W. E., Kleijnen, J. P., Van Beers, W. C., Van Nieuwenhuyse, I. (2007). "Kriging metamodeling in constrained simulation optimization: an explorative study," in 2007 Winter Simulation Conference, pp. 355-362: IEEE.

Dengiz, B., Alabas-Uslu, C., Dengiz, O. (2009). "A tabu search algorithm for the training of neural networks," *Journal of the Operational Research Society*, 60 (2): 282-291.

Jones, D. R., Schonlau, M., Welch, W. J. (1998). "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, 13(4): 455-492.

Moghaddam, S., Mahlooji, H. (2017). "A new metamodel-based method for solving semi-expensive simulation optimization problems," *Communications in Statistics-Simulation and Computation*, 46 (6): 4795-4811.

Mohammad Nezhad, A., Mahlooji, H. (2014). "An artificial neural network meta-model for constrained simulation optimization," *Journal of the Operational Research Society*, 65 (8): 1232-1244.

Molga, M., Smutnicki, C. (2005). "Test functions for optimization needs," https://www.robertmarks.org/Classes/ENGR5358/Papers/functions.pdf

Nguyen, A.-T., Reiter, S., Rigo, P. (2014). "A review on simulation-based optimization methods applied to building performance analysis," *Applied Energy*, 113: 1043-1058.

Østergård, T., Jensen, R. L., Maagaard, S. E. (2018). "A comparison of six metamodeling techniques applied to building performance simulations," *Applied Energy*, 211: 89-103.

Rigoni, E., Lovison, A. (2007). "Automatic sizing of neural networks for function approximation," in 2007 IEEE International Conference on Systems, Man and Cybernetics, pp. 2005-2010: IEEE.

Shan, S., Wang, G. G. (2010). "Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions," *Structural and Multidisciplinary Optimization*, 41 (2): 219-241.

Shi, L., Ólafsson, S. (2000). "Nested partitions method for global optimization," *Operations Research*, 48 (3): 390-407.

Tsai, S. C., Fu, S. Y. (2014). "Genetic-algorithm-based simulation optimization considering a single stochastic constraint," *European Journal of Operational Research*, 236 (1): 113-125.

Vicario, G., Craparotta, G., Pistone, G. (2016). "Meta-models in computer experiments: Kriging versus artificial neural networks," *Quality and Reliability Engineering International*, 32 (6): 2055-2065.

Wang, L. (2005). "A hybrid genetic algorithm–neural network strategy for simulation optimization," *Applied Mathematics and Computation*, 170 (2): 1329-1343.

Wang, G. G., Shan, S. (2007). "Review of metamodeling techniques in support of engineering design optimization," *Journal of Mechanical Design*, 129 (4): 370-380.

Wang, X., Wang, G. G., Song, B., Wang, P., Wang, Y. (2019). "A Novel evolutionary sampling assisted optimization method for high dimensional expensive problems," *IEEE Transactions on Evolutionary Computation*, DOI: 10.1109/TEVC.2019.2890818.

Willmes, L., Back, T., Jin, Y., Sendhoff, B. (2003). "Comparing neural networks and kriging for fitness approximation in evolutionary optimization," in the 2003 Congress on Evolutionary Computation, CEC'03., vol. 1, pp. 663-670: IEEE.

Xu, J., Huang, E., Chen, C.-H., Lee, L. H. (2015). "Simulation optimization: A review and exploration in the new era of cloud computing and big data," *Asia-Pacific Journal of Operational Research*, 32 (3): 1550019.

Yuan, R., Guangchen, B. (2009). "Comparison of neural network and Kriging method for creating simulation-optimization metamodels," in 2009 eighth IEEE international conference on dependable, autonomic and secure computing, 2009, pp. 815-821: IEEE.