Chapter 5

Metamodel-Based Simulation Optimization: A Systematic Literature Review

Chapter details

Chapter DOI:

https://doi.org/10.4322/978-65-86503-88-3.c05

Chapter suggested citation / reference style:

Amaral, João V. S., et al. (2022). "Metamodel-Based Simulation Optimization: A Systematic Literature Review". *In* Jorge, Ariosto B., et al. (Eds.) *Uncertainty Modeling: Fundamental Concepts and Models*, Vol. III, UnB, Brasilia, DF, Brazil, pp. 119–171. Book series in Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity.

P.S.: DOI may be included at the end of citation, for completeness.

Book details

Book: Uncertainty Modeling: Fundamental Concepts and Models

Edited by: Jorge, Ariosto B., Anflor, Carla T. M., Gomes, Guilherme F., & Carneiro, Sergio H. S.

Volume III of Book Series in:

Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity

Published by: UnB City: Brasilia, DF, Brazil Year: 2022

DOI: https://doi.org/10.4322/978-65-86503-88-3

Metamodel-Based Simulation Optimization: A Systematic Literature Review

João Victor Soares do Amaral[1*], José Arnaldo Barra Montevechi[1], and Rafael de Carvalho Miranda[1]

[1]Industrial Engineering and Management Institute, Federal University of Itajubá, Brazil.

E-mail addresses: joao.amaral@unifei.edu.br (J.V.S. Amaral), montevechi@unifei.com.br (J.A.B. Montevechi), rafael.miranda@unifei.edu.br (R.C. Miranda),

*Corresponding author

Abstract

Over the past few decades, metamodeling tools have received attention for their ability to represent and improve complex systems. The use of metamodeling techniques in optimization problems via simulation has grown considerably in recent years to promote more robust and agile decision-making, determining the best scenario among the solution space to the problem. In this sense, this Chapter discusses the state of the art of metamodeling-based simulation optimization, presenting the gaps, opportunities, and future perspectives found in literature.

Keywords: Metamodeling; Simulation; Optimization; Machine Learning.

1. Introduction

Production systems need to include increasingly effective decision-making tools to ensure organizational competitiveness, customer satisfaction, and cost reductions (Salam and Khan, 2016). According to Law (2013), analytical solutions are effective when systems are relatively simple. However, the author states that most real systems are too complex for analytical analyses. Simulations were developed to predict and evaluate complex, stochastic, and often analytically intractable system performance (Xu et al., 2016).

One advantage of using simulations is reduced decision-making risk, since simulations allow different scenarios to be evaluated without physically interfering in production systems (Helleno et al., 2015). Since not all possible scenarios can be practically evaluated, Simulation Optimization (SO) is used to achieve these goals (Barton, 2009).

The SO field has progressed significantly over the past decade with newly developed algorithms, implementations, and applications. There have been significant contributions to this field of research given the work of researchers and practitioners, mainly in the fields of engineering, industrial operations, mathematical programming, statistics, machine learning, and computer science (Amaran et al., 2016).

SO finds the input values of a simulation model that gives the best outputs (Belgin, 2019). According to Fu (2002), SO can be divided into two phases: 1) generating possible candidates for solving the problem; and 2) evaluating solutions using simulation model estimates. Figure 1 shows this process.

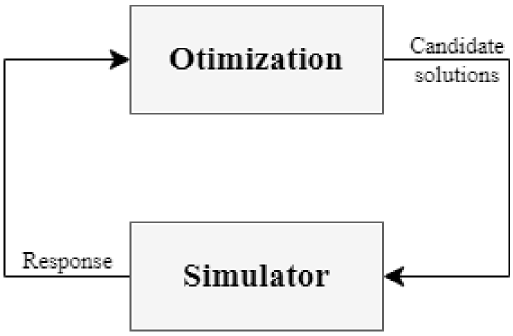

Figure 1: General SO process.
Source: Adapted from Fu (2002).

According to Miranda et al. (2017), several SO techniques have been developed in recent years, e.g., heuristics and metaheuristics, that have still been little explored by scientific literature.

SO methods can be classified into two categories, model-based and metamodel-based methods. Moghaddam and Mahlooji (2017), explain that the optimization and simulation modules are directly integrated with each other in model-based methods, where optimization occurs according to the solutions that need to be evaluated, then the simulation evaluates them. A third component, called a metamodel, exists in metamodel-based approaches, which identifies and estimates the relationship between simulation model inputs and outputs as a deterministic mathematical function that is used to evaluate possible simulation and optimization solutions (Barton and Meckesheimer, 2006; Moghaddam and Mahlooji, 2017; Parnianifard et al., 2020a).

The model-based approach is the most popular and widespread method according to literature. However, when a simulation model is very complex, and when a problem solution space is large, a SO process using these techniques is too slow, and may require hours or months to converge to an acceptable local optimum value, making use unfeasible for several applications (Cai et al., 2019; Miranda et al., 2014). On the other hand, metamodeling techniques have been increasingly used over the last decade to reduce computational optimization times (Parnianifard et al., 2019).

Metamodeling techniques have been applied in several engineering and management branches, e.g., optimally allocating operators and equipment on factory floors (Amaral et al., 2022), planning and controlling traffic to minimize travel times (Xiao Chen, Osorio, and Santos, 2019), structurally designing vehicles to reduce weight (Fu and Sahin, 2004), finding optimal sizes for concrete barrier designs (Yin et al., 2016), and geometrically optimizing laminated composite wind turbine blades to reduce weight (Albanesi et al., 2018), etc.

Currently, several algorithms are used to build metamodels. Some examples include Response Surface Methodology (RSM) (Hassrnaayebi et al., 2019; Shirazi; Mahdavi; Amiri, 2011), the *Kriging* algorithm (Baquela and Olivera, 2019; Pedrielli et al., 2020), the Radial Basis Function (RBF) (Christelis and Mantoglou, 2016; Yin et al., 2016), and Artificial Neural Networks (ANN) (Raei et al., 2019; Storti et al., 2019).

Given these various methods, and the growing importance of metamodeling for complex engineering problems, this study seeks to provide readers with a review of the main metamodeling concepts as applied to SO problems. Furthermore, this chapter will conduct a Systematic Literature Review (SLR), to provide a holistic view of how this topic is addressed in literature.

2. Background

According to Lin and Chen (2015), optimization via simulation is a powerful decision-making tool, since this technique can understand various relationships in complex real-world systems, to identify the best solution to a given problem. According to Souza Junior et al. (2019), SO helps decision-makers invest or allocate resources in new or existing production systems.

Amaran et al. (2016), define SO as an optimized Objective Function (OF), that is subject to constraints, which is then evaluated using stochastic simulations. Carson and Maria (1997), state that SO is a process of identifying the best input values for simulated system variables, and evaluating solutions using a loop created between the optimization algorithm and the simulation model.

The SO field of study has evolved significantly over the last decade, in light of newly developed algorithms, software programs, and applications. Amaran et al. (2016), state that the general formulation of an SO problem can be expressed as in Eq. (1).

$$\begin{gathered} \mathrm{Min}\ \mathbb{E}_{\omega}[f(x,y,\omega)] \\ s.a.\ \ \mathbb{E}_{\omega}[g(x,y,\omega)] \leq 0 \\ h(x,y) \leq 0 \\ x_l \leq x \leq x_u \\ x\ \in\ \mathbb{R}^n, y\ \in\ \mathbb{D}^m \end{gathered} \tag{1}$$

Where $\mathbb{E}_{\omega}$ is the expected value of function *f*, which is evaluated by simulating continuous *x* inputs, or discrete *y* inputs, subjected to a random number vector ω. Likewise, restrictions are defined by function *g* values that are evaluated in each simulation. Problems may also contain other constraints (represented by *h*) that do not involve random variables, or contain constraints linked to decision variables.

By contrast, Miranda et al. (2017), states that long computational times are required for the optimization algorithm to converge to a good result for SO problems addressing complex systems with very large solution spaces. Barton (2009), states that this has led researchers to develop specialized SO methods, which include:

- ranking and selection;
- heuristics and metaheuristics;
- random searches; and
- metamodeling.

According to De La Fuente and Smith (2017), the more complex the studied system, the more time required for performing optimizations. According to the authors, to obtain good results in a timely manner, one would need to elevate problems to another level of abstraction, i.e., a metamodel.

Metamodeling is commonly used for developing representations of simulation models (De La Fuente et al., 2019). A metamodel can understand relationships between decision variable values and simulation outputs, providing OF approximations faster than the original simulation model (Jalali and Nieuwenhuyse, 2015). In layman's terms, a simulation model is an abstraction of a real system, and similarly, a metamodel is an abstraction of a simulation model (a model of another model), making evaluations simpler and faster, albeit while less accurately representing real systems. Figure 2 shows this relationship.

Once a metamodel has been trained and validated it can replace the original simulation model in SO processes, thereby considerably reducing computational times for optimization convergence, as shown in Figure 3. This process is called Metamodel Based Simulation Optimization (MBSO).

Several metamodeling methods and applications have emerged in literature in recent years to create more reliable metamodels. Barton (2009), stands out, especially for comparing the main methods used for these applications, i.e., RSM, RBF, Kriging, RNA, and spline. Hannah, Powell, and Dunson (2014), proposed an MBSO method based on a semi-convex regression. Yousefi et al. (2018), used ensemble methods in an agent-based simulation model to reduce average wait times in the healthcare sector,

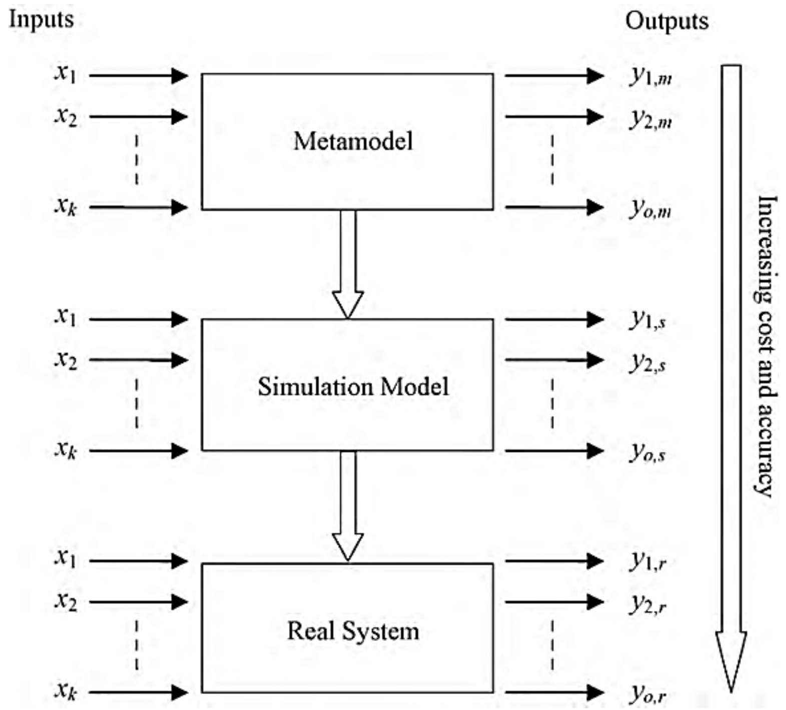

Figure 2: Relationship between real-world scenarios, simulation models, and metamodels.
Source: (Li et al., 2010).

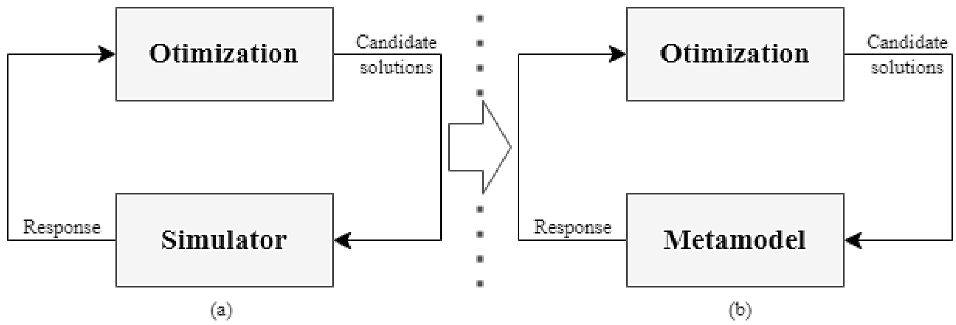

Figure 3: SO process (a) and MBSO process (b).

Souza Junior et al. (2019), studied the integrations between metamodeling, parallel processing and metaheuristics appltied to production engineering problems, and Woldemariam and Lemu (2019), used ANN and Genetic Algorithm (GA) metamodeling to optimize fluid dynamics analysis models. Hullen et al. (2020), proposed strategies for dealing with SO uncertainty by combining ML with stochastic programming, robust optimizations, and discrepancy modeling.

Sections 2.1 – 2.3 address the concepts involved in the main machine learning algorithms used to build metamodels, performance metrics, hyperparameter optimization techniques used in these algorithms, and a brief explanation of one of the most widely-used MBSO designs, the Latin Hypercube Design (LHD), all to aid in better understanding the algorithms and techniques common to metamodeling studies.

2.1 Machine Learning Algorithms

2.1.1 Gaussian Process

Also known as the Kriging algorithm, the Gaussian Process (GP) is a non-parametric Bayesian approach using variables following Gaussian distribution (Kleijnen, 2014). The main goal is to define a probability for infinite functions, by adding finite points that result in a function passing directly over these points (Ostergard et al., 2018). Nguyen-Tuong et al. (2009), state that given a set of n training data $\{x_i, y_i\}_{i=1}^{n}$, the function $f(x_i)$ seeks to learn from input data (x_i) transformed into response variables (y_i), converging according to Eq. (2), where ϵ_i is the Gaussian noise function, with zero mean, and variance σ_n^2.

$$y_i = f(x_i) + \epsilon_i \tag{2}$$

The GP is a stochastic process, whereby any finite group of model entry points are incorporated into Gaussian distribution, expressed as $f(x) \sim GP[\mu(x), K(X, X))]$, where $\mu(x)$ and $K(X, X)$ are the mean and the covariance functions, respectively (Wan and Ren, 2015). Here, the covariance functions are normally given via a Gaussian kernel, as per Eq. (3) (Nguyen-Tuong et al., 2009).

$$k(x_p, x_q) = \sigma_s^2 \exp\left(-\frac{1}{2}(x_p - x_q)^T W(x_p - x_q)\right) \tag{3}$$

Where $\sigma_s^2$and W are the variance and amplitude of the Gaussian kernel. Furthermore, Eq. (4) shows the joint distribution for the estimated ($f(x_*)$) and observed (y) values, given a given input x_*. By conditioning the joint distribution, one can extract the predicted mean value ($f(x_*)$), and the associated variance ($V(x_*)$), as per Eq. (5) and Eq. (6) (Nguyen-Tuong et al., 2009).

$$\begin{bmatrix} y \\ f(x_*) \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) + \sigma_*^2 & k(X, x_*) \\ k(x_*, X) & k(x_*, x_*) \end{bmatrix}\right) \tag{4}$$

$$f(x_*) = k_*^T (K + \sigma_*^2 I)^{-1} y = k_*^T \alpha \tag{5}$$

$$V(x_*) = k(x_*, x_*) - k_*^T (K + \sigma_*^2 I)^{-1} k_* \tag{6}$$

Where $k_* = k(X, x_*), K = K(X, X)$, and α are the SO prediction vectors.

2.1.2 Polynomial Regression

PR is a collection of statistical and mathematical techniques that can be applied to develop, improve, and optimize processes. It has been successfully used in recent decades to represent stochastic systems (Parnianifard et al., 2020a). PR was developed over 50 years ago for exploration and extrapolation applications (Barton and Meckesheimer, 2006), and according to Parnianifard et al. (2019), literature cites PR as being used to:

1. approximate relationships between dependent and independent variables;
2. investigate and determine the best process operating conditions; and
3. implement more robust responses.

According to Myers, Montgomery, and Anderson-Cook (2016), PR for a given response is based on Taylor series expansion theory to a data set. The general formulation for linear regression (first-order PR) is given by Eq. (7) (Parnianifard et al., 2019).

$$y = f(X) = \hat{\beta}_0 + \sum_{i=1}^{k} \hat{\beta}_i x_i + \varepsilon \tag{7}$$

Where k is the number of independent variables. If more complex response surfaces cannot be represented by linear PR, quadratic (second-order) PR can be used. Its general formulation is given in Eq. (8) (Dengiz and Belgin, 2015; Osorio and Bierlaire, 2013; Parnianifard et al., 2019; Parnianifard et al., 2020a).

$$y = f(X) = \hat{\beta}_0 + \sum_{i=1}^{k} \hat{\beta}_i x_i + \sum_{i=1}^{k} \hat{\beta}_{ii} x_i^2 + \sum_{i=1}^{k} \sum_{i<i'=2}^{k} \hat{\beta}_{ii'} x_i x_{i'} + \varepsilon \tag{8}$$

Where $\hat{\beta}_0, \hat{\beta}_i, \hat{\beta}_{ii}$ and $\hat{\beta}_{ii'}$ are the unknown coefficients, and ε is the model noise. The number of expressions p is $p = k + 1$ for linear regression, $p = \frac{1}{2}(k+1)(k+2)$ for quadratic regression, and $p = \frac{1}{6}(k+1)(k+2)(k+3)$ for cubic regression (Parnianifard et al., 2019). One of the main advantages of PR is the model interpretability and adaptability to different data types. However, Ostergard et al. (2018), state that these can be computationally inefficient compared to more sophisticated high dimension expansions, e.g., GP and SVM.

2.1.3 Artificial Neural Networks

ANNs have been widely applied to classification, regression, and time-series problems. They consist of an input layer, one or more hidden layers, and an output layer. (Hassoun, 1995). The general ANN and neuron structures are shown in Figures 4 and 5, respectively.

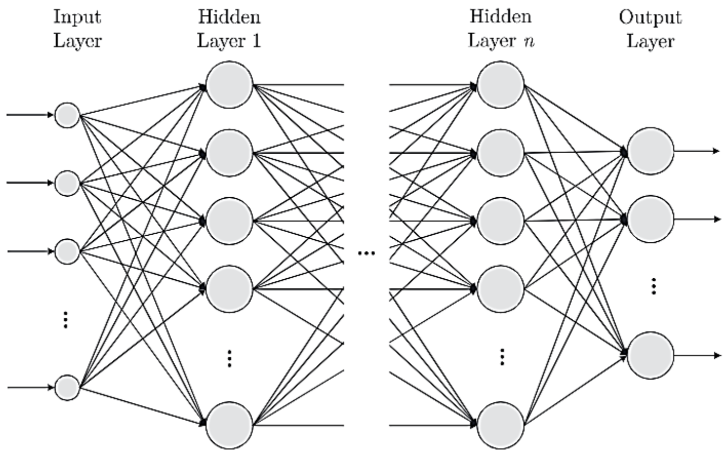

Figure 4 - Structure of an Artificial Neural network
Source: Adapted from Storti et al. (2019)

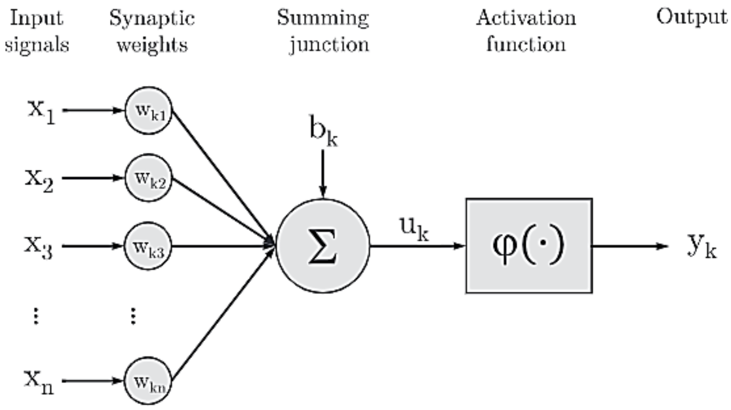

Figure 5 - General structure of a Neuron
Source: Adapted from Storti et al. (2019)

According to Li et al. (2010), information transmission occurs via neuron connections through layers. Also according to the authors, a typical ANN can be represented with three layers and one neuron in the output layer, as per the following formula:

$$\hat{y} = \hat{f}(X) = \sum_{j=1}^{j} w_j f\left(\sum_{i=1}^{k} v_{ij} f(x_i) + a_j\right) + \beta + \varepsilon \tag{9}$$

Where:

- $\boldsymbol{X}$ is a k dimension vector;
- $f(.)$ is the transfer function;
- ε the random error with mean 0;
- v_{ij} represents the connection weights between the i neurons in the input layer and the j neurons in the hidden layer;
- a_j is the bias of the j^{th} hidden layer neuron;
- w_j is the weight of connections between the j^{th} hidden neuron and the output neuron;
- j is the total number of neurons in the hidden layer; and
- β is the output neuron bias (Li et al., 2010).

According to Oliveira et al. (2010), weights and biases are adjusted using supervised training algorithms. The back-propagation algorithm is most widely used, where the model is fed with standard inputs, and the signal is propagated to the output layer to generate predictions. This error then adjusts the output neuron, hidden neuron, and input neuron connection weights.

2.1.4 Random Forest

Breiman (2001), introduced the Random Forest (RF) method, which combines many randomly uncorrelated decision trees to obtain predictions based on average votes among all individuals (Li et al., 2020). RF is one of the most efficient ML algorithms in terms of accuracy and execution speed for large databases. It is also a simple method, and its variables and results can be easily interpreted (Crisci et al., 2012).

Dai et al. (2018), state that RF is a combination of decision trees $\{h(x, \Theta_k), k = 1, 2, \dots\}$, where each tree receives a random vector $\{\Theta_k\}$. Outputs are numerical, and training data are independently and identically distributed in regression problems (Breiman, 2001). The root mean square error associated with each tree *h(x)* is defined by Eq. (10), for the X vector and Y output (Dai et al., 2018).

$$E_{X,Y}(Y - h(\boldsymbol{X}))^2 \tag{10}$$

The procedure for adjusting the RF is simple. The first step involves adjusting all *m* independent trees. Then a sample size *N* is obtained at each interaction, and *k* attributes are randomly selected to generate new uncorrelated trees (De La Fuente and Smith, 2017). The individual predictions of each m^{th} tree are guided by a set of *if-then-else* rules, characterized by Θ (Hastie et al., 2008). According to Breiman (2001), an FR forecast is calculated using the average from the *k* trees *h(x)*, belonging to the forest, as per Eq. (11).

$$f(X) = \frac{1}{M}\sum_{m=1}^{M} T(\boldsymbol{X}: \Theta_m) \tag{11}$$

Where *M* is the number of trees, $T(\boldsymbol{X}:\Theta_m)$ is the model for a single tree, *X* is the training data vector, and Θ_m is the set of rules for the trees.

2.1.5 Gradient Boosted Trees

The Boosting method was initially proposed by Schapire (1990), to build highly accurate models based on low learning converging algorithms, or decision trees for Gradient Boosted Trees (GBT). Friedman (2002), proposed a variation of this method called Stochastic Gradient Boosting, introducing randomness to the procedure proposed by Schapire (1990). Friedman (2002), states that the trees are trained using a random sample $\tilde{N}$ of the training data at each iteration of the algorithm without replacements, so $\tilde{N} < N$, where *N* is the size of the training data. The GBT algorithm differs from bagging methods, since the first seeks to sequentially build models (trees), while the second is adjusted based on models built in parallel (Xia et al., 2017).

This algorithm is based on sets of decision trees. A new tree is generated at each interaction based on the residues of previous trees (De La Fuente and Smith, 2017). Given a dataset $\{x_i; y_i\}$ size N, input attributes are represented by x_i, while y_i is the response variable. The objective of the algorithm is to determine the function $F^*(x)$ that expresses the relationship between x and y to minimize the loss function as per Eq. (12). This relationship is additively approximated in Eq. (13) (Xia et al., 2017).

$$F^*(x) = \arg min_f \; \Psi\,(y, F(x)) \tag{12}$$

$$F_m^*(x) = \sum_{m=0}^{M} \beta_m h(x; \Theta_m) \tag{13}$$

The ideal base learners ($h(x;\Theta_m)$) are represented by $\beta_m h(x;\Theta_m)$, where β_m and Θ_m are optimal coefficients and parameters for the base learner, respectively. Frieman (2002), presents a pseudocode for GBT as described in Algorithm 1.

Algorithm 1: Stochastic Gradient Boosted Trees

1 $F_0(x) = arg\,min_\gamma \; \Psi\,(y_i, \gamma)$

2 $For\ m = 1\ to\ M\ do$:

3 $\{\pi(i)\}_1^N = rand_{perm}\,\{i\}_1^N$

4 $\tilde{y}_{\pi(i)m} = -\left[\frac{\partial \Psi\left((y_{\pi(i)}, F(x_{\pi(i)})\right)}{\partial F(x_{\pi(i)})}\right]_{F(x)=F_{m-1}(x)} \quad , i = 1, \tilde{N}$

5 $\{R_{lm}\}_1^N = L - terminal\ node\ tree(\{\tilde{y}_{\pi(i)m}, x_{\pi(i)}\}_1^{\tilde{N}})$

6 $\gamma_{lm} = arg\,min_\gamma \sum_{x_{\pi(i)} \in R_{lm}} \Psi\left(y_{\pi(i)}, F_{m-1}(x_{\pi(i)}) + \gamma\right)$

7 $F_m(x) = F_{m-1}(x) + \nu \,.\, \gamma_{lm}\, l(x \in R_{lm})$

8 *End For.*

Source: Friedman (2002).

Where $\Psi\,(y_i,\gamma)$ represents the loss function, R_{lm} represents each region formed by the l nodes of the m^{th} trees, and $\pi(i)\}_1^N$represents integer (1, ... N) permutations for training data $\{y_i, x_i\}_1^N$.

Line (4) in the algorithm gives the gradient of the loss function, or the pseudo-residues of the last iteration, and γ_{lm} is the set of optimal expansion coefficients corresponding to each tree node in the m^{th} iteration.

2.1.6 Support Vector Machine

A Support Vector Machine (SVM) from Vapnik (1999), is one of the most popular ML methods proposed in recent years (Cao et al., 2020). It was initially built to classify problems but has been widely used for regression problems (Cao et al., 2020, Hastie et al., 2008).

For classification problems, the method seeks boundaries to separate training data into respective classes with the least amount of error. A hyperplane is defined based on vectors built from observations located at boundaries, so data on different sides of the hyperplane are allocated in different classes (Ostergard et al., 2018).

Jeng (2006), explains that SVM tries to find a maximum margin between two classes using linear vector combinations for these types of problems, and this is solved using quadratic programming with linear constraints. According to the authors, this algorithm can be extended to solve nonlinear regression problems by introducing the $\mathcal{E}$ parameter, which represents the loss function. Li et al. (2010), gave a mathematical formulation of SVM as per Eq. (14):

$$\hat{y} = \hat{f}(X) = w\phi(X) + b \tag{14}$$

Where $X \in \Re^k$ is the k-dimensional input, ϕ is a nonlinear transformation from $\Re^k$ to $\Re^h$, with $h > k$, and $\hat{f}(X)$ describes the hyperplane in $\Re^h$. The authors emphasize that coefficients *w* and *b* can be obtained by minimizing the risk function, as per Eq. (15).

$$R_{svm}(C) = C\sum_{i=1}^{n} C\big(\hat{f}(X_i), y_i\big) + \frac{1}{2}\|w\|^2 \tag{15}$$

Where:

$$C\big(\hat{f}(X_i), y_i\big) = \begin{cases} \left|\hat{f}(X) - y\right| - \mathcal{E} & se \left|\hat{f}(X) - y\right| \geq \mathcal{E} \\ 0 & Contrary\ Case \end{cases} \tag{16}$$

The final SVM method prediction function is obtained after solving the minimization problem via Eq. (17) (Vapnik, 1999).

$$f(X, a_i, a_i^*) = \sum_{i=1}^{n} (a_i - a_i^*)K(X, X_i) + b \tag{17}$$

Where *X* is the new model input, $K(X, X_i)$ is the Kernian function representing the distance between *X* and X_i, and a_i, a_i^* and *b* are coefficients obtained by minimizing the risk function (Li et al., 2010).

Oliveira et al. (2010), state that SVM efficiency depends on correctly choosing the hyper-parameters *C*, epsilon ($\mathcal{E}$) and Kernel (K). According to Cherkassky and Ma (2004), the ideal *C* value of can be calculate via Eq. (18):

$$C = \max\left(\left|\bar{y} + 3\sigma_y\right|, \left|\bar{y} - 3\sigma_y\right|\right) \tag{18}$$

Where $\bar{y}$ and σ_y are the mean and standard deviations of the *y* values in the training data. To calculate $\mathcal{E}$, the authors recommend making a first pass with the algorithm using the default parameters to calculate the prediction error δ associated with the database, where $\delta = \hat{y} - y_i, (i = 1, \ldots, n)$.

$$\mathcal{E} = 3\sigma\sqrt{\frac{\ln n}{n}} \tag{19}$$

Where σ is the standard deviation of error δ, and *n* is the number of database examples.

2.2 Performance Metrics

An important step in building metamodels is evaluating them to choose the model that best fits the data, and aid in selecting and/or optimizing the parameters. Bergmeir and Benítez (2012), stated that the Mean Square Error (MSE) or Root MSE (RMSE), the Mean Absolute Error (MAE), and the Mean Absolute Percentage Error (MAPE), were the main error metrics cited in literature. According to Hu et al. (2006), the coefficient of correlation (R^2) was another important regression metric.

Taking y_i as the observed (actual) value at point *i*, $\hat{y}_i$ as the predicted value, and *n* as the size of the test data, RMSE, MAE and MAPE can be calculated as per Eq. (20), Eq. (21), and Eq. (22), respectively (Bergmeir and Benitez, 2012; Chai and Draxler, 2014).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{20}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{21}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|100\ \frac{y_i - \hat{y}_i}{y_i}\right| \tag{22}$$

There is no consensus as to the model accuracy values for metamodel evaluation metrics. However, one could state that a metamodel with $R^2 > 0.90$ is reasonable, $R^2 > 0.95$ is accurate, and $R^2 > 0.99$ is almost perfect (Ostergard et al., 2018).

R^2 is an accuracy metric with an amplitude that varies from 0 to 1. As shown in Eq. (23), it is calculated by dividing the sum of the squares of the residuals $(y_i - \hat{y}_i)^2$ by the total sum of the squared residuals $(y_i - \bar{y}_i)^2$. High R^2 values correspond to good model and training data fits, but do not guarantee reduced errors (Hu et al., 2006).

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \tag{23}$$

Where:

$$\bar{y}_i = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{24}$$

2.3 Hyper-Parameter Optimization

Choosing adequate hyper-parameters for ML algorithms is a fundamental step for increasing the learning capacity of these algorithms. Given that these parameters cannot be manually defined, one must use optimization techniques to find the best performing configuration extraction from the algorithm (Xia et al., 2017).

Currently, there are several optimization methods for selecting hyper-parameters, like AG, ant colony, grid search, Bayesian optimization, etc. (Hullen et al., 2020; Kim; Boukouvala, 2020; Li et al., 2019). In our study, we chose to use GA given its execution speed, its ability to adapt to different variable types, and given that the initial values do not need to be defined to perform optimization. Table 1 gives a brief comparison of the main hyper-parameter optimization methods.

Table 1 - Comparison of common methods for hyper-parameter optimization.

Optimization Methods	**Strengths**	**Weaknesses**	**Complexity (time)**
Grid Search	Simple	Takes a lot of time and is efficient only for categorical variables	n^k
Randon Search	Is more efficient than the Grid Search	Does not consider previous results and is not efficient with conditional variables	n
Gradient-based	Allows for parallelism	Supports only continuous variables and detects only large locations	n^k
Bayesian GP	Has fast convergence for continuous variables	Low capacity for parallelism and is not efficient for conditional variables	n^3
Bayesian TPE	Has fast convergence for continuous variables	Low capacity for parallelism	*nlogn*
Sequential model-based algorithm configuration	Is efficient for all variable types	Low capacity for parallelism	*nlogn*
Hyperband	Maintains conditional dependencies	Not efficient for conditional variables and requires small subsets to be relevant	*nlogn*
Bayesian hyperBand	Is efficient for all variable types	Requires small subsets to be representative	*nlogn*
Genetic algorithm	Allows for parallelism	Low capacity for parallelism	n^2
Particle swarm	Is efficient for all variables types Allows for parallelism	Requires proper start up	*nlong*

k is the number of hyperparameters, and *n* is the number of hyperparameter values.

Source: Adapted from Yang and Shami (2020).

A Genetic Algorithm (GA) is a heuristic technique that was inspired by the theory of natural selection. It was initially proposed by De Jong (1975), and Holland (1975), and is based on Darwin's theory of the survival of the fittest. It can be successfully applied to linear and nonlinear programming problems (Bashiri and Geranmayeh 2011; Lessmann et al., 2006). The search space in GA comprises candidate solutions, each represented by a chromosome (Alabas et al., 2002). Each chromosome comprises genes, representing encoded parameters for a candidate solution, and aptitude, or OF, to reach the best solution for the problem. The objective of GA is to combine genes to obtain new chromosomes with better OF values (Li et al., 2010).

The GA comprises three basic operators, i.e., reproduction, crossover, and mutation. Reproduction is a process by which individual parameters, or genes, are copied based on their OF values. Individuals with higher OF values are more likely to generate one or more offspring in the next generation. When a parameter is reproduced it creates an exact replica of itself. Then, the crossover operator randomly combines newly produced parameters. Finally, mutations generate changes to parameter values, playing a secondary role, because mutation rates are generally low (Alabas et al., 2002; Storti et al., 2019).

GA applied to hyper-parameter optimization was studied by Hou et al. (2019), and Hullen et al. (2020). This method evaluates several possible parameter combinations $\{a_1, ..., a_i\}$ of the ML algorithm. The cross-validation technique is used to calculate the error associated with each combination. k-fold cross-validation is one of the most important techniques for evaluating regression and classification methods. It divides training data both equally and randomly into k parts, using k-1 parts for training the algorithm, and the remainder for testing. The error of each array is calculated k times, and a different part $k_i^{'}$ is selected for testing at each iteration, with $i = \{1, ..., k\}$. Then, performance is measured based on averaged errors in k validation interactions, which is a more robust metric than if it were measured using just one observation (Bergmeir and Benítez, 2012). Roy and Datta (2019), suggested using 10-fold cross-validation.

2.4 Design of Experiments

According to Montevechi et al. (2012), experiments are a series of tests with purposeful changes made to input variables to identify and analyze the effects of these changes to response variables within the system. According to the authors, the Design of Experiments (DOE), is a set of methods for designing and conducting experiments. Several DOE techniques have been cited in literature, e.g., orthogonal arrays (Miranda et al., 2017), Central Composite Designs (CCD), (Villarreal-Marroquín et al., 2013), factorial designs (Dengiz et al., 2006), etc.

According to the Structured Literature Review (SLR) in Section 4, the most common DOE method for metamodeling projects is the Latin Hypercube Design (LHD). According to Parnianifard et al. (2020), LHD is a computational DOE technique based on space-filling. For n input variables, m sample points are randomly obtained within m equally probable intervals. Figure 6 gives an example of an LHD array with n=2 decision variables, and a m=7 sample size.

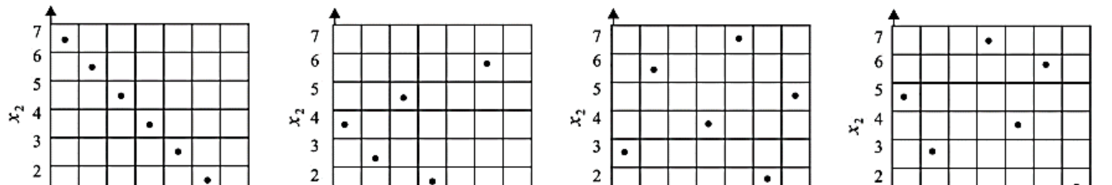

Figure 6: Examples of LHD arrays with 2 input variables and 7 sample points Source: Adapted from Luo, Ji and Lu (2019)

According to Parnianifard et al. (2020), the LHD procedures are:

- The value ranges for each variable are divided into *m* subranges of equally probable magnitudes. The number of sampling points *m* is defined by the researcher. In general, it should be greater than the total points produced by the CCD.
- The second step is allocating the *m* points between each lower and upper interval limit, with each point (1, 2, ..., *m*) in one row and column of the sample space.

3. Systematic Literature Review

As shown in the previous section, there is a wide range of methodologies and applications surrounding MBSO. Sections 3 and 4 provide a Systematic Literature Review on the topic to provide the reader with an overview of how metamodeling has been approached both in academia and in practice.

3.1 Literature Reviews on Metamodeling Simulation Optimization

We conducted a search in the Scopus and Web of Science scientific databases in July 2020 to look for studies including Literature Reviews (LR) on MBSO, or other related topics. In this step we searched for terms related to metamodeling (metamodel, metamodeling, meta-model, or machine learning), optimization via simulation (Simulation optimization, simulation-based optimization, or optimization via simulation) using Boolean logic (AND/OR), and Literature Reviews (literature reviews, the state-of-the-arts, overviews, or surveys), with a filter for documents in English.

12 LRs on topics related to MBSO were returned, however, these focused on a certain stages of metamodeling projects, one or a few metamodeling algorithms or types of simulations, and none contained complete discussions on MBSO knowledge that should be used by academics and practitioners for developing projects. This section contributes by offering a comprehensive view of MBSO using the scientific rigor of SLR (Y. Xiao and Watson 2019), and CIMO (Context, Interventions, Mechanisms, Outcomes) (Denyer et al., 2008).

The first study returned was published from the Winter Simulation Conference in 1986, addressing the frequency domain experimentation technique and RSM for dealing with discrete variable problems (Schruben, 1986). At the same conference, twenty-three years after the publication of the study by Schruben (1986), Barton (2009) gave an overview of SED metamodeling techniques, describing the main algorithms used in metamodeling according to literature i.e., RSM, RNA, Kriging, spline, RBF, and provided a framework for conducting metamodeling projects using local or global searches. That same year, Kleijnen (2009), published a review of Kriging metamodeling and its association with techniques like bootstrapping, DOE, sensitivity analysis, and optimization.

In 2010, a study systematically compared five SO metamodeling algorithms (i.e., RNA, RBF, Kriging, SVM, and multivariate regression splines) for decision support systems (LI et al., 2010). In 2014, a study was carried out on Kriging metamodeling with parametric and distribution-free bootstrapping applied to optimization problems via deterministic and/or stochastic simulations (Kleijnen 2014). That same year, Figueira and Almada-Lobo (2014), studied hybrid SO method classifications and taxonomies, explaining questions on procedures and methods related to metamodeling.

In 2015, Dellino et al. (2015), described robust optimization concepts using Kriging metamodeling in a chapter of their book, applying these to an economic order quantity inventory model. A survey was carried out again in 2015 on the practical use of search experimentation techniques, i.e., SO, DOE, and metamodeling applied in SED (Hoad et al., 2015). Two years later, an LR was conducted presenting a general review of several metamodeling algorithms (i.e., polynomial regression, Kriging, RBF, SVM, and mixed metamodels), DOE techniques (i.e., geometrical, statistical, and optimal designs), and mechanisms (i.e., expected improvements, efficient global optimizations, bumpiness, etc.), for problems containing continuous variables (Vu et al. 2017). That same year, two other LRs addressed metamodeling using Kriging and polynomial regression (Kleijnen 2017a, 2017b).

Finally, only one study used a SLR to study metamodeling for robust black-box SO subjected to uncertainties (Parnianifard et al. 2019). These studies served as the basis for conducting our SLR on MBSO. We should mention that our SLR did not seek to conduct an exhaustive review of literature on MBSO, but rather to provide academics and practitioners in the area with a broad view of the subject by consolidating and explaining existing MBSO projects, while also defining the frontier for knowledge in this area, by identifying and presenting current opportunities and challenges related to this topic.

3.2 The Systematic Literature Review Method

A SLR is a methodology for outlining the theoretical-scientific basis necessary for understanding a topic. This is done by collecting, understanding, synthesizing, and evaluating a set of scientific studies (Levy and Ellis, 2006; Xiao and Watson, 2019). According to Sousa Junior et al. (2019), SLRs assist researchers in performing methodological steps, e.g., defining problems, selecting methods, collecting data, and performing analysis, to mitigate flaws in the conclusions of studies, thereby offering greater clarity on employed research procedures.

SLR is not the same as a traditional LR, but rather seeks to clearly explore a specific issue by consulting existing literature. An SLR differs from other LR methodologies given its exact principle, whereby researchers must pre-specify and clarify their inclusion/exclusion criteria to readers (Denyer and Tranfield, 2009). Oliveira et al. (2016), point out that one of the advantages of SLR relative to traditional RL methods, is that SLRs avoid data bias and distortions. A SLR mainly seeks to promote advancements and to develop new studies, given the fact that a new frontier of knowledge can only be established after defining the current frontier.

3.3 Employing the Systematic Literature Review Method

Our SLR adopted the same procedures as Oliveira et al. (2016), and Sousa Junior et al. (2019), to systematically investigate MBSO knowledge, by identifying borders and gaps in existing literature, thereby providing new guidelines for future studies. This was done in four main stages, i.e., planning, researching/sorting, analyzing/synthesizing, and presenting the results.

3.3.1 Planning

This step was developed by a panel of experts in the area to better investigate this field of research. The panel included 3 Ph.Ds with academic and practical experience in MBSO projects, in addition to a Ph.D. and M.Sc. researching the same topic.

The study was inspired given the expressive growth in MBSO publications, the differing applications to knowledge areas, and given the lack of studies comprehensively and systematically synthesizing and discussing knowledge on this subject. We expect that our results can aid all project stages within this area, and present the state-of-the-arts for MBSO by compiling existing studies, and mainly proposing future opportunities for additional research.

The main SLR objectives are:

- identify and explore the main MBSO aspects in studies;
- analyze and summarize the results;
- discuss the results, define the state-of-the-arts, and any research gaps and trends that could be further explored.

To do this, Research Questions (RQ) need to be defined to extract and analyze intrinsic knowledge on the topic. Several frameworks and methods for structuring RQs have been addressed in literature e.g., PICO, PICOS or PICOC (Population, Interventions, Comparisons, Outcomes, and Context) (Kloda et al. 2020), SPICE (Settings, Perspectives, Interventions, Comparisons, and Evaluations) (Booth, 2006), SPIDER (Samples, Phenomenon of Interest, Designs, Evaluations, and Research Type) (Methley et al. 2014), ECLIPSE (Expectations, Clients, Location, Impacts, Professionals, and Services) (Wildridge and Bell 2002), etc.

We used the CIMO-logic framework as proposed by Denyer et al. (2008), which combines problem Contexts requiring Interventions, generated using a Mechanism, in order to produce Outcomes. This approach has been used by several authors to study engineering problems (Aouadni et al., 2019; Balan, 2018; Makhashen et al., 2020; Costa et al., 2020; Ilk et al., 2020; Kochan and Nowicki, 2018; Tanila et al., 2020). The RQs in our study were divided into the four elements outlined in the CIMO-logic method:

- context: real or theoretical problems...
- interventions: ... using metamodeling...
- mechanisms: ...combined with optimization techniques via simulations...
- outcomes: ...to find the best performing solution within the project scope and constraints.

The RQs were formulated in compliance with these objectives, within the scope of this study, and to identify existing gaps in literature. The RQs are:

- RQ1) What is the role of metamodeling in SO projects, and what are its main applications? (Context)
- RQ2) What are the main simulation and optimization techniques used with metamodeling? (Mechanisms)
- RQ3) What are the most widely-used metamodeling algorithms in MBSO? (Interventions)

- RQ4) What sampling strategies are normally used? (Interventions)
- RQ5) How are results measured/validated? (Outcomes)

The information extracted from the articles was divided into two categories to answer the RQs, as shown in Figure 7.

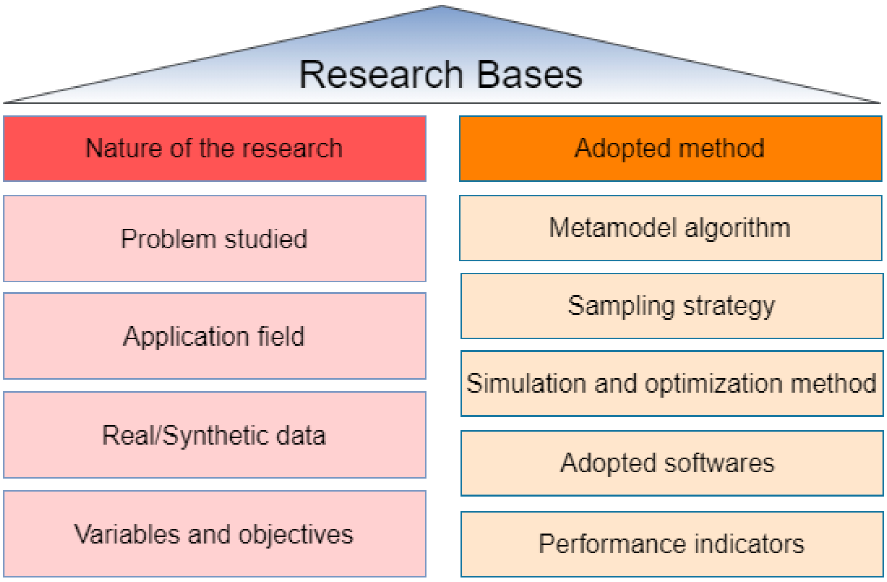

Figure 7: Research pillars and article extracted data

The first category investigates the nature of the work relative to the area of industry that the studies focus on, what types of problems are addressed, whether the problem is real or theoretical (based on data from literature), and how decision variables and problem objectives are framed. The second category deals with method-related issues e.g., simulation type, the metamodeling algorithms used, the search engines used in the optimizations, sampling strategies, software programs, and how results were measured. Finally, the main MBSO frameworks from literature will be presented, along with two new frameworks that complement the main points found in the SLR.

3.3.2 Researching and Sorting the Articles

We conducted exploratory research in the Scopus and Web of Science (WoS) scientific databases on the topic, since these databases are the main data citation sources (Mongeon and Paul-Hus 2016), and are representative of publications on the subject (AGHAEI et al., 2013). Thus, terms related to metamodeling (metamodel, metamodeling, meta-model, and machine learning), and SO (Simulation optimization, simulation-based optimization, optimization via simulation, and SO) were searched for in both databases using Boolean Logic (AND/OR) in the process. The meta-search returned 188 articles from the WoS and 293 articles from Scopus, resulting in 481 studies.

The initial article selection phase for downloading was made by reading the titles/abstracts of the articles that met the following inclusion criteria:

a) contained SO metamodeling search terms in titles/abstracts/keywords;
b) published in peer-reviewed journals;
c) written in English; and
d) had full versions available for download.

209 studies were excluded from the search after applying the inclusion criteria for failing to meet at least one of the previously defined criteria. 3 (0.6%) did not have the search terms in the titles, abstracts, or keywords, 203 (42, 2%) were not published in peer-reviewed journals, 5 (0.8%) were not written in English, and 1 was not available for download. 269 articles were downloaded in total for quality assessment, 144 from Scopus, and 125 from the WoS. Then 104 articles were removed due to duplications.

Quality assessments consist of reading the titles, abstracts, objectives, methods, results, and discussions. 21 articles were excluded from analysis at this stage for the following reasons, 11 (52.3%) mentioned "machine learning" but did not use ML or metamodeling, 8 (38.1%) used metamodeling/ML for purposes other than SO, and 2 (9.5%) did not use simulations. 143 articles were selected for full reading after the quality assessment for information extraction and data analysis. Figure 8 shows the search/screening procedures.

Searching/screening procediments			
Databases selection	**Scopus**	**WoS**	**Total**
Search the literature	n = 293	n = 188	n = 481
Screen for inclusion	n = 144	n = 125	n = 165
Assess quality	n = 126	n = 121	n = 143

Figure 8: Searching/Sorting

Figure 8 shows the results of the search/sort phase. 293 articles were downloaded from Scopus, and 188 from WoS after the search phase. 144 articles from Scopus and 21 from WoS were included for screening. 104 eligible articles from WoS were excluded because they were duplicates with Scopus. Finally, 126 studies from Scopus and 17 from WoS were selected for analysis and data extraction in the quality assessment phase.

3.3.3 Analysis and Synthesis

The articles were evaluated in 9 items to answer the RQs, as shown in Figure 7, corresponding to:

1. define the problem;
2. industrial sector;
3. data source;
4. variables and objective types;
5. metamodeling algorithms;
6. sampling strategies;
7. simulation and optimization methods;
8. software used;
9. performance indicators and validation methods.

Similar nomenclature standardizations were performed after data extraction, e.g., the gaussian process and Kriging were understood as being the same algorithm, and the Latin Hypercube Design and Latin Hypercube Sampling refer to the same DOE technique.

Descriptive data, table, and graph statistics were made after the synthesis using the Software R program. The results were analyzed in line with the RQs related to the three aforementioned categories, serving as a basis for the results and conclusions that aided us in developing a body of evidence on the main MBSO perspectives discussed in this SLR.

3.3.4 Results Presentation

The results were summarized in graphs and tables to discuss the topic, revealing compelling SLR insights. We reflected on and identified the gaps and perceptions on the development of this theme within the scientific community based on the analysis of the interactions between SO and metamodeling.

The main results are summarized in Tables 2 - 6 below following the analyses and syntheses, wherein all RQs will be answered, and where we discuss the state-of-the-arts for MBSO, presenting practices observed in literature, and possibilities for future research on this topic. The 143 articles were divided into two categories according to the problem origin, e.g., real or theoretical, to elucidate the relationship between academia and practical problems facing industry/service providers.

4. Results and SLR Discussion

This section presents the data obtained from the articles and answers the RQs, in the order following the pillar arrangements, i.e., nature of research, methods, and state-of-the-arts.

4.1 Nature of research

This section discusses the “nature of research” pillar, which is directly linked to RQ1 (“What is the role of MBSO, and what are its main applications?”). The main problem-related aspects and industrial sectors, economic sectors, and problem specificities like objectives and sizes studied in literature are presented. Table 2 shows the problem type, and industrial and economic sectors.

The objects of study in the articles were classified and grouped into more than 18 problem classes. Given the wide variety of problems, only the six most frequent classes are shown in Table 2 i.e., Process Control, Stock Control, Product Design, Resource Allocation, Mathematical Models, Traffic Models, together constituting more than 65% of the problems found. The most studied issue was using MBSO for control processes to optimize production parameters for certain goods. Most works (74.1%), addressed this problem type relative to managing water resources to find well-water injection and extraction pumping rates for supplying demand while reducing energy costs, wear and tear, and watershed contamination e.g., (Ataie-Ashtiani et al. 2014; Candelieri et al., 2018; Hussain et al., 2015; Roy and Datta 2019a; Timani and Peralta 2017). Other applications were determining optimal parameters for plastic injection processes (Dang 2014; Villarreal-Marroquín et al., 2011, 2013), cutting parameters for semiconductor wires (Monostori and Viharos 2001), and sheet metal pressing (WANG et al., 2018).

Table 2 - Nature of the research.

Problem Type	**Real**	**Theoretical**	**Total**	**Cum. %**
Process Control	16 (10.4%)	10 (07.2%)	26 (17.2%)	17.2
Stock Control	01 (00.6%)	20 (13.1%)	21 (13.7%)	31.2
Product Design	08 (05.2%)	08 (05.2%)	16 (10.5%)	41.6
Resource Allocations	11 (07.2%)	05 (03.3%)	16 (10.5%)	52.1
Mathematical Models	00 (00.0%)	11 (07.2%)	11 (07.2%)	59.3
Traffic Models	08 (05.2%)	01 (00.6%)	09 (5.88%)	65.1
Industrial Sector				
Not Specified	03 (02.3%)	36 (27.9%)	39 (30.2%)	30.2
Water Management	19 (14.7%)	10 (07.7%)	29 (22.5%)	52.7
Urban Traffic	18 (13.9%)	02 (01.5%)	20 (15.5%)	68.2
Energy Generations / Distribution	06 (04.6%)	01 (00.8%)	07 (05.4%)	73.6
Health care	06 (04.6%)	00 (00.0%)	06 (03.9%)	78.3
Pharmaceutical / Chemical Industry	03 (02.3%)	02 (1.5%)	05 (03.9%)	82.2
Economic Sector				
Not Specified	03 (02.3%)	36 (27.9%)	39 (30.2%)	30.2
Primary	19 (14.7%)	10 (07.7%)	29 (22.4%)	52.7
Secondary	15 (11.6%)	07 (05.4%)	22 (17.0%)	69.8
Terciary	33 (25.5%)	06 (04.6%)	39 (30.2%)	100.0

The second problem class was related to inventory controls and management activities, e.g., sizing the number of kanbans for shop floors e.g., (Alabas et al., 2002; Dengiz; Alabas-Uslu and Dengiz, 2009; Hurrion, 1997; Huyet and Paris, 2001), and lot size (Sousa Junior et al., 2019; Greenwood et al., 1998; Hachicha, 2011; Horng and Lin, 2017; Jackson et al., 2019; Moghaddam and Mahlooji, 2016, 2017; Nezhad and Mahlooji, 2014; Parnianifard et al., 2020b; Zakerifar et al., 2011). Both dealt with theoretical cases, where data sources came from other articles, or from classic problems in specialized literature. In our SLR, only one article addressed the stock control problem by studying a real case using metamodeling via Kriging in discrete SO to optimize emergency fuel stocks in ships (Quan et al. 2013).

Many authors have also studied MBSO techniques to optimize product designs. Examples include applications related to the automotive (Fu and Sahin, 2004; Ivanova and Kuhnt, 2014; Stork et al., 2008), energy (Sharif and Hammad 2019; Storti et al., 2019), hydropower (Bananmah et al., 2020; Mooselu et al., 2019), and transit infrastructure (Yin et al., 2016) sectors. Many authors have also studied MBSO applied to resource allocation problems (e.g., (Coelho and Pinto 2018; Song et al., 2005; Yousefi and Yousefi 2019; Zeinali et al., 2015), to synthetic mathematical functions (e.g., Baquela and Olivera, 2019; Kim and Boukouvala, 2020; Gonzalez et al., 2020; Wang et al., 2020), and to urban traffic issues (Chen et al., 2019; Chong and Osorio 2018; Osorio and Bierlaire 2013; Osorio and Chong 2015).

Another important factor is that theoretical studies and real cases are balanced in both of these problem classes, and in most of other classes found in the SLR, indicating the interest and development in the area, in academia, and in practice. However, it is worth mentioning that studies focused on theoretical cases for some problem classes, e.g., inventory controls, and we recommend exploring this problem more in real contexts, since inventory management is a strategic activity for most organizations.

The second aspect studied was the industrial sector of focus in the studies. 52.0% of the articles focused on water resource management, energy generation/distribution, urban traffic planning, healthcare, and the chemical/pharmaceutical industry. This shows that most MBSO efforts are focused on essential goods and services production, providing cost reductions and improvements that benefit the greatest number of people. About 30.2% of all studies did not specify the industrial sector. It is noteworthy that among the real cases, only 4.3% did not specify the industrial sector.

This is positive because more precise problem indications help future researchers and practitioners select the methodologies that best fit their industrial sector. Regarding the economic sector of the studies, primary, secondary, and tertiary sectors were equally represented, characterizing the breadth and versatility of MBSO techniques.
Table 3 shows the variables and objectives of the studies.

Table 3 - Problem variables and objectives.

Variables	**Real**	**Theoretical**	**Total**	**Cum. %**
Nº of Variables (median)	10	5	6	-
Continuous	34 (26.7%)	37 (29.1%)	71 (55.9%)	55.9
Discreet	12 (09.4%)	23 (19.1%)	35 (27.6%)	83.5
Hybrid	07 (05.5%)	07 (05.5%)	14 (11.0%)	94.5
Binary	02 (01.6%)	01 (00.8%)	03 (02.4%)	96.9
Not Specified	00 (00.0%)	03 (02.4%)	03 (02.4%)	99.2
Categorical	01 (00.8%)	00 (00.0%)	01 (00.8%)	100.0
Optimization Objectives				
Single Objective	53 (35.6%)	54 (36.2%)	107 (71.8%)	71.8
Multi Objective	18 (12.1%)	22 (14.8%)	40 (25.8%)	98.7
Not Specified	01 (00.7%)	01 (00.7%)	02 (01.3%)	100.0

Regarding the problem sizes studied, real case applications had a median of 10 variables, compared to 5 variables for theoretical cases, as shown in Table 3. Furthermore, many authors have studied problems with more than 100 variables (e.g., (Chen et al., 2019; Chong and Osorio 2018; Jackson et al., 2019; Osorio 2019b; Osorio and Punzo 2019; Timani and Peralta 2017), extending up to 16,200 variables for a mesoscopic simulation model calibration problem for controlling traffic (Osorio 2019a). Regarding the type of variable, 83.5% of the problems had continuous or discrete variables. The first type had infinite solution spaces, and the second had delimited solution spaces, with an average of 203,963 possible scenarios, ranging from 36 to $1x10^{62}$ scenarios, with a 90% percentile at $2.5x10^{20}$.

Although most studies addressed single-objective problems, 25.8% studied metamodeling for multi-objective optimization problems, mainly dealing with identifying Pareto Frontiers. We can observe the importance of using metamodeling, since recursive SO methods are practically unfeasible for dimensionality and solution space problems found in this SLR, in addition to proving to be good choices for multi objective optimizations, which are a class of NP-hard problems (Baquela and Olivera 2019).

We also analyzed metamodeling use versus model-based techniques. Although 61.3% of all studies did not specify the reason for choosing metamodeling over other methods, 33.9% of all articles used metamodeling to reduce project optimization times, while other methods could not converge with adequate computation time given restricted simulation budgets (e.g., Parnianifard et al., 2018; Song et al., 2005; Villarreal-Marroquín et al., 2013). Furthermore, 3.0% used metamodeling to provide quick answers to operational problems requiring short-term decision-making time frames (e.g., Dunke and Nickel, 2020; Calahorrano et al., 2016). Metamodeling was also used to perform joint optimizations taking data from several models (e.g., Bartz-Beielstein et al., 2018), to quickly generate entire solution spaces (e.g., Stork et al., 2008), and better understand optimization processes (e.g., Huyet and Paris, 2001).

We can answer RQ1 ("What is the role of metamodeling in SO projects, and what are its main applications?"), according to the information in Tables 2 and 3, and in the discussion. Metamodeling is used to reduce computational times for problems where solution spaces and model complexity make more traditional recursive methods too expensive or unfeasible. Furthermore, recursive methods are unfeasible depending on the decision-making time frame, in light of the advent of industry 4.0, and the use of simulations as decision support tools in real or near real time, meaning that metamodels need to be used. Once trained and validated, they can generate good results in mere seconds.

Several authors have used this strategy for operational support systems (e.g., Dunke and Nickel, 2020; Calahorrano et al., 2016; Pedrielli et al., 2020; Shirazi et al., 2011; Steer et al., 2012). Regarding the main MBSO applications in real cases, process control problems, resource allocation and sizing, urban traffic controls, and product designs are particularly worth mentioning. Furthermore, there was fair distribution among different economic sectors, allowing us to infer that MBSO is a broad method that can generate good results when facing various problems, including NP-hard problems.

4.2 Methods used

This section presents the most widely-used methods, according to literature, for metamodeling algorithms, search methods (optimization), simulation types, sampling strategies, software programs, and programming languages. Tables 4 and 5 summarize the results, showing the six most frequent technologies/methods for each studied topic.

We noticed that the number of occurrences exceeds the total number of articles analyzed for certain topics due to the fact that some studies used more than one method, either by comparing or combining them. For example, Li et al. (2010), compared metamodels using RNA, RBF, SVM, Kriging, and MARS for a resource allocation problem, and Bartz-Beielstein et al. (2018), combined metamodels using linear regression, RF, and Kriging via an ensemble model, etc. (e.g., Lal and Datta, 2020; Parnianifard et al., 2020a; Ranjbar et al., 2020).

The first aspect in Table 4 was the type of simulation used by the researchers. 25% of the studies did not state the type of simulation used, and this only appeared as a recommendation for future studies so researchers can identify the type of simulation applicable to their specific problems in literature, and identify which tools can be used and associated with it. Of all the articles, 26.3% used SED. According to Law (2013), SED is defined as a system modeling and simulation technique that evolves over time, where events change state at discrete points in time.

Several authors have used SED for problems related to the secondary and tertiary economic sectors, like inventory controls (e.g., Dengiz et al., 2009; Quan et al., 2015), shop floor planning (e.g., Dunke and Nickel, 2020; Greinacher et al., 2020), allocating and sizing resources (Aydin et al., 2018; Coelho and Pinto 2018; Cuckler et al., 2017), and production scheduling (Chang, 2016; Hassannayebi et al., 2019). No SED metamodeling application was conducted for the primary economic sector, perhaps because this type of simulation is commonly taught in specific engineering courses, limiting this tool's dissemination. We can therefore state that there is a possible literature gap that future researchers could explore, i.e., using metamodeling and SED to optimize processes in the primary sector.

Table 4 - Method used.

Type of Simulation	**Real**	**Theoretical**	**Total**	**Cum. %**
SED	14 (09.7%)	24 (16.6%)	38 (26.3%)	26.4
Not Specified	12 (08.3%)	24 (16.6%)	36 (25.0%)	51.4
Finite Element Method	13 (09.0%)	13 (09.0%)	26 (18.0%)	69.4
Simulating Traffic	16 (11.1%)	1 (00.7%)	17 (11.8%)	81.3
Monte Carlo Simulations	01 (00.7%)	5 (03.5%)	06 (04.2%)	85.4
Finite Volume Method	04 (02.8%)	00 (00.0%)	04 (02.8%)	88.2
Optimization Method				
Genetic Algorithm	12 (13.1%)	10 (10.9%)	22 (24.1%)	24.2
Trust Region	11 (12.0%)	01 (01.1%)	12 (13.1%)	37.4
NSGAII	08 (08.8%)	02 (02.2%)	10 (10.9%)	48.4
Particle swarm	06 (06.6%)	03 (03.3%)	09 (09.9%)	58.2
Simulated annealing	03 (03.3%)	02 (02.2%)	05 (05.5%)	63.7
Metamodeling Algorithm				
Kriging	22 (09.0%)	31 (12.7%)	53 (21.7%)	21.7
Polynomial Regression	20 (08.2%)	19 (07.8%)	39 (16.0%)	37.7
RNA	26 (10.7%)	13 (05.3%)	39 (16.0%)	53.7
Ensembles	06 (02.5%)	15 (05.1%)	21 (08.6%)	62.3
RBF	06 (02.5%)	09 (03.7%)	15 (06.1%)	68.4
SVM	02 (00.8%)	08 (03.3%)	10 (04.1%)	71.5
Sample Strategy				
Latin hypercube design	23 (15.2%)	38 (25.1%)	61 (40.3%)	40.4
Random	10 (06.6%)	10 (06.6%)	20 (13.2%)	53.6
Not Specified	10 (06.6%)	03 (02.0%)	13 (08.6%)	62.3
Factorial design	04 (02.6%)	06 (04.0%)	10 (06.6%)	68.9
Central composite design	03 (02.0%)	03 (02.0%)	06 (04.0%)	72.8
Orthogonal array	01 (00.7%)	04 (02.6%)	05 (03.3%)	76.2

By contrast, the second most widely-used simulation type was numerical simulation using the finite element method (18.0%), and most applications were focused on the primary economic sector (50.0%), for optimizing pumping rates for water extractions (e.g., Dhar and Datta, 2009; Hussain et al., 2015; Roy and Datta, 2018a, 2019b). Traffic simulations were conducted for 11.8% of all articles, corresponding to microscopic models (Guo and Zhang 2014; Osorio and Bierlaire, 2013; Osorio and Nanduri, 2015), mesoscopic models (Chen et al., 2019; Chen et al., 2016; Osorio, 2019a), and macroscopic models (Chen et al., 2014; Zhang et al., 2017). Metamodeling acts as a means for minimizing transit model error in this type of simulation by optimizing the model's input parameters (e.g., origin-destination matrix), minimizing the average travel time by optimizing traffic signal cycle times, defining public transportation and/or toll fees, etc.

Several other types of simulations were also identified, but were less-popular than those mentioned before. These include Monte Carlo simulations (e.g., Song et al., 2005; Wade, 2019), the finite volume method (e.g., Mooselu et al., 2019; Storti et al., 2019), the differential elements method (e.g., Alizadeh et al., 2017; Li et al., 2020), and agent-based simulations (e.g., Yousefi and Yousefi, 2019; Zeinali et al., 2015), etc.

The second item shown in Table 4 was the optimization method used in the studies. It is worth mentioning that metamodels are not optimization methods, but rather simplified models of original simulations. Researchers need to apply appropriate search methods to metamodels to perform optimizations. Many optimization algorithms were found in the studies. Genetic Algorithms were most-commonly used, especially for single-objective network planning problems (Shariatinasab et al., 2008; Vahidi et al., 2008; Ye and You, 2015), for resource allocation (Sousa Junior et al., 2020; Yousefi et al., 2018), process control (Ali et al., 2018; Luo et al., 2019; Ouyang et al., 2017), and product designs (Albanesi et al., 2018; Storti et al., 2019).

The second most frequently used search method was the trust region method, which is mostly used with quadratic polynomial metamodels for urban traffic control problems (Chen et al., 2019; Chong and Osorio, 2018; Osorio and Bierlaire, 2013; Osorio and Chong, 2015; Osorio and Nanduri, 2015), traffic model calibrations (Li et al., 2017; Osorio 2019b), emergency evacuations (Lv et al. 2015), and network planning (Ye and You 2015). The third most commonly applied search method was the Non-dominated Sorting Genetic Algorithm II (NSGA-II), an evolutionary multi-objective optimization algorithm, which was used mostly for optimizations, especially optimizations for process control problems. (Alizadeh et al., 2017; Dhar and Datta, 2009; Hussain et al., 2015; Ranjbar et al., 2020; Sreekanth and Datta 2011), and for product designs (Mooselu et al., 2019; Sharif and Hammad 2019)

Other commonly used methods in literature were particle swarm optimizations (e.g., Li et al., 2020; Yadav et al., 2016), and simulated annealing (e.g., Christelis and Mantoglou, 2016; Ciuffo and Punzo, 2014). By stratifying the main types of simulations, we see that Genetic Algorithms were used most in SED, the finite element method, and the finite volume method. The trust-region was most used for traffic simulations, and Monte Carlo simulations used mostly Gold Region, Ranking, and Selection methods.

With respect to RQ2 ("What are the main simulation and optimization techniques used in conjunction with metamodeling?"), we can conclude that SED is the most popular simulation technique considering the total number of publications, and the most popular technique for secondary economic sector applications, given that most industrial processes have independent and identically distributed variables, and can be modeled as discrete systems. The finite element method was most common for the primary economic sector, led by studies on process controls for water extraction activities. Traffic simulations, in particular microscopic simulations, for urban traffic planning and management, were most frequently used as simulation techniques in the tertiary economic sector. Furthermore, researchers used Genetic Algorithms as a search engine in single-objective optimization processes more frequently, and the NSGA II algorithm more frequently for multi-objective optimizations.

The third analyzed item was the type of metamodeling algorithm used. Table 4 shows that the most frequent algorithm was Kriging, appearing in 53 articles (21.7%). Kriging was initially proposed by Daniel G. Krige (1951), for geostatistical applications. It is a highly flexible interpolation method, given its ability to employ a wide variety of correlation functions (Parnianifard et al., 2018). Kriging has been applied to a wide variety of problems because of this flexibility, e.g., for controlling inventories (Azizi, Seifi, and Moghadam 2019; Quan et al., 2013; Smew, Young, and Geraghty 2013), product designs (Bartz-Beielstein et al., 2018; Ivanova and Kuhnt, 2014; Yaohui 2017), process controls (Boukouvala and Ierapetritou, 2013; Candelieri et al., 2018; Roy and Datta 2018b), etc.

PR is another class of classic algorithms found in literature (16%). PR is an explicit mathematical expression, that is either linear or nonlinear, that aims to approximate relationships between independent and dependent variables (Parnianifard et al., 2019). PR is commonly found in literature as RSM (Aydin et al., 2018; Cuckler et al. 2017; Hassannayebi et al., 2019), as quadratic polynomials (Xiao et al., 2015; Zlatinov and Laskowski, 2015), and as first-orders (Hong et al., 2013; Villarreal-Marroquin et al., 2013). Many authors have also used polynomials to approximate true functions in local searches via the trust region method (e.g., Chen et al., 2019; Chen et al., 2014; Chong and Osorio, 2018; Osorio and Chong, 2015; Osorio and Nanduri, 2015).

ANNs were the third most frequently used algorithm. These algorithms are some of the most important artificial intelligence techniques, given their ability in solving extremely difficult problems that do not have any known analytical solutions. ANNs comprise both interconnected and adaptive elements, which respond to stimuli, similar to the human nervous system (Hachicha, 2011). Choosing the appropriate architecture is a crucial success factor for ANN applications, however, standard formula cannot be determined, and each case must be analyzed individually.

Hyperparameter selection techniques used in metamodels are important for these types of problems, and trial and error methods (Ataie-Ashtiani et al., 2014; Sreekanth and Datta, 2011), or more efficient optimization methods, like Genetic Algorithms for selecting hyper-parameters, can be used (Hullen et al., 2020; Kim et al., 2016).

Several authors (4th position) also used ensemble methods, or algorithms that combined several metamodels, be them similar or not, to formulate metamodels with greater learning capacities, like bagging (Chen et al., 2017; Yousefi et al., 2018), stacking (Bartz-Beielstein et al., 2018), AdaBoost (Yousefi et al., 2018; Yousefi and Yousefi, 2019), etc.

Algorithms using decision tree-based ensembles were the most frequent, and RF use was common (e.g., Bartz-Beielstein et al., 2018; Candelieri et al., 2018; Sousa Junior et al., 2019; Sousa Junior et al., 2020; Steer et al., 2012), along with GBT (e.g., (Sousa Junior et al., 2019; Sousa Junior et al., 2020). Many applications deal with using RBF algorithms (Sousa Junior et al., 2019; Sousa Junior et al., 2020). e.g., Ali et al., 2018; Christelis and Mantoglou, 2016; Horng and Lin, 2018; Krityakierne et al., 2016; Stork et al., 2008), and SVM (e.g., Hou et al., 2019; Lal and Datta, 2020; Li et al., 2019; Mirenderesgi and Mousavi, 2016).

Based on the above data, we can answer RQ3 (“What are the most commonly used metamodeling algorithms in MBSO?”). The main algorithms are Kriging, Polynomial Regression (especially RSM), ANNs, ensemble methods (in special RF and GBT), RBF, and SVM. The most common algorithms stratified by type of simulation in SED are Polynomial Regressions, Kriging, ANNs, and decision trees. For the finite element method, they were Kriging, ANNs, Polynomial Regressions, MARS. and RBF. 80% of all traffic simulations used either Polynomial Regressions or Kriging. Monte Carlo simulations and the finite volume method mostly used ANNs. Table 5 shows the main hyper-parameters of these algorithms and their frequencies .

The last item in Table 4 was the sampling strategy used by the researchers to build the metamodel, addressing RQ4 ("Which sampling strategies were used?"). One of the critical success factors in metamodel use is the training database. So researchers must choose the correct sampling method that can reduce the number of simulations needed for modeling, while also representing the search space well (Vu et al. 2017). 61 articles (40.3%) used the DOE technique, known as LHD, followed by random sampling (13.2%), Factorial Design (6.6%), CCD (4.0%), the Orthogonal Array (3.1%), etc., in addition to another 13 articles (8.6%), that either did not use, or did not specify, the sampling technique.

LHD was first proposed by McKay et al. (1979), and is a very popular stratified sampling method for metamodeling studies. Given *n* samples and *s* variables, LHD divides each variable's region of into *n* disjointed intervals of equal probability. Then the sample space is selected by building a *n x s* matrix. The columns are randomly selected from a permutation of [*1*, ..., *n*], and each row corresponds to a hyper-rectangle cell. After building the matrix, a point from each cell is sampled (Luo et al., 2019).

In addition to good representative samples in the solution space, the sample size in LHD can be defined by the user. This can lead to success in building metamodels that, unlike techniques like the orthogonal array or CCD, have greater decision-making power, in function of trade-offs between experimental performance time and metamodel accuracy. Although the sample size can be defined by the user, the academic consensus is that it should be at least 10 times greater than the number of problem variables (Kim and Boukouvala, 2020; Parnianifard et al., 2018; Parnianifard and Azfanizam, 2020; Smew et al., 2013). We should note that LHD was the most used simulation type, except for in traffic simulations, which used random sampling more frequently.

Furthermore, 80 articles (61.5% of the studies using some sampling technique) used the fixed sampling strategy, i.e., only a set of samples size *w* is taken at the beginning of the project, and the metamodel is trained with this set alone. For the incremental strategy (38.5%), an initial data sample size ξ is taken, and the metamodel is trained and optimized with the precision and region direction results containing the optimum level, adding a new sample set size δ_i to the database. Then the metamodel is retrained with base size $\xi_{i+1} = \xi_i + \delta_i$, where $i \in \mathbb{Z}$ is the number of algorithm iterations.

This algorithm, which is responsible for recursive and adaptive learning, is referred to as Efficient Global Optimization (EGO) in most studies. The choice for the point(s) that should be added to each interaction is usually made using the Expected Improvement (EI) algorithm (Coelho and Pinto 2018; Pedrielli et al., 2020; Quan et al., 2013; Rojas-Gonzalez et al., 2020; Wang et al. 2018). Table 6 shows the main software programs used, and performance metrics found in literature.

Regarding the type of technology used in the studies, we observed from the first topic in Table 6 that most authors (19.7%), did not specify the modeling and simulation software they used, while 14.9% used programming languages like C++ (e.g., Song et al., 2011), Python (e.g., (Sousa Junior et al., 2019; Kim and Boukouvala, 2020), or Matlab (e.g., Parnianifard and Azfanizam, 2020; Rojas-Gonzalez et al., 2020). Regarding proprietary software used for simulations, we can cite the use of Arena®, one of the first software programs developed with 2D animation for modeling discrete systems. FEMWATER and MoldFlow were the most common software programs for the finite element method, along with Aimsun for traffic simulations.

Table 5 – The main metamodeling algorithm hyper-parameters.

Algorithm	Amplitude	Most Frequent
Kriging		
Correlation Function	{Gaussian, polynomial, exponential}	Gaussian
Lenghscale (θ)	$[10^{-10}; 10^{10}]$	10
Epsilon[1]	$[10^{-6}; 10^{-5}]$	-
Geometric Tolerance[1]	[0; 0.3]	-
Max Nº. of vectors[1]	[38; 47]	-
RP		
Degree	{1º, 2º, 3º, 4º, 5º}	2º
RNA		
Transfer function	{linear, sigmoid, quadratic error, hyperbolic tangent, rectified linear unit, RBF}	Linear
Training function	{backpropagation, levenberg-marquardt, feedforward}	Backpropagation
Learning rate	[0.01; 0.4]	0.1
Epochs	[200; 1500]	1500
Nº of layers	[1; 4]	1
Nº of neurons	[1; 66]	4
Momentum	0.1	0.1
RF		
Nº of trees	[10; 50]	10
Maximum Depth	2	2
Nº of variables per node[3]	int(p/3) – 1, int(p/3), int(p/3) + 1	-
GBT		
Nº of trees	[1; 100]	100
Learning rate2	[0.05; 0.30]	-
Min. Obs. per leaf2	[0.12; 0.50]	-
RBF		
Function	{gaussian, splines, multiquadratics, cubic}	Gaussian
Scale (sigma)	[1; 20]	1
Regularization	$[10^{-10}; 10^{10}]$	-
SVM		
C[4]	$[\lvert\bar{y} - 3\sigma\rvert;\ \lvert\bar{y} + 3\sigma\rvert]$	20.73
Epsilon[4]	$\left[0; 3\sigma\sqrt{\ln n/n}\right]$	0.0779
Kernel	{RBF, Gaussian}	Gaussian
Gama	[0.0224; 0.1294]	0.0673

[1]Hyper-parameter cited but not quantified in the SLR, so the hyper-parameter as suggested by Rostami, Khaksar, and Manshad (2014) was used.
[2]Hyper-parameter cited but not quantified in the SLR, therefore, the hyper-parameter as suggested by Ganjisaffar, Caruana, and Lopes (2011) was used.
[3]Hyper-parameter cited but not quantified in SLR articles, however, this was soon adopted as suggested by Dai et al. (2018), where *p* is the number of decision variables in the problem.
[4]The C and epsilon values should be calculated from an algorithm first run using the default parameters, where $\bar{y}$ and σ_y are the mean and standard deviations of the *y* values in the training data, *n* is the sample size, and σ is the variance of the difference between the estimated and observed values. For further details see Oliveira et al. (2010).

Table 6 - Software and metrics.

Simulation Software	Real	Theoretical	Total	Cum. %
Not Specified	05 (03.4%)	24 (16.3%)	29 (19.7%)	19.7
Programming Language	07 (04.7%)	15 (10.2%)	22 (15.0%)	34.7
Arena	04 (02.7%)	08 (05.4%)	12 (08.1%)	42.9
FEMWATER	04 (02.7%)	04 (02.7%)	08 (05.4%)	48.3
Aimsun	07 (04.7%)	00 (00.0%)	07 (04.7%)	53.1
MoldFlow	02 (01.3%)	02 (01.4%)	04 (02.7%)	55.8
Metamodeling Software				
Matlab	26 (18.5%)	26 (18.5%)	52 (37.1%)	37.1
Not Specified	24 (17.1%)	24 (17.1%)	48 (34.2%)	71.4
Python	03 (02.1%)	03 (02.1%)	06 (04.2%)	75.7
R	03 (02.1%)	03 (02.1%)	06 (04.2%)	80.0
GAMS	03 (02.1%)	03 (02.1%)	06 (04.2%)	84.3
Minitab	02 (01.4%)	02 (01.4%)	04 (02.8%)	87.1
Performance Metrics				
Mean square error (MSE)/Root MSE	32 (12.6%)	36 (14.1%)	68 (26.8%)	26.8
Coefficient of Correlation (R, R^2, R_{adj})	24 (09.4%)	26 (10.2%)	50 (19.7%)	46.5
Not Specified	24 (09.4%)	13 (05.1%)	37 (14.6%)	61.0
Mean absolute percentage error (MAPE)	14 (05.5%)	12 (04.7%)	26 (10.2%)	71.3
Mean absolute error (MAE)	05 (01.9%)	07 (02.7%)	12 (04.7%)	76.0
Standardized residuals error (SRE)	02 (00.7%)	05 (02.0%)	07 (02.7%)	78.7
Validation Technique				
Not Specified	28 (22.6%)	20 (16.1%)	48 (38.7%)	38.7
Train/test split	17 (13.7%)	11 (08.9%)	28 (22.6%)	61.3
Leave one out cross-validation	09 (07.3%)	11 (08.9%)	20 (16.1%)	77.4
k-fold cross-validation	06 (04.8%)	11 (08.9%)	17 (13.7%)	91.1
Does not use validation	00 (00.0%)	04 (03.2%)	04 (03.2%)	94.4
New test data	02 (01.6%)	02 (01.6%)	04 (03.2%)	97.6

Regarding the software or programming language used in developing the metamodels and the optimization structure, Matlab was used by 37.1% of all researchers. However, Python and R, which are open-source languages ,were widely used in the field of Data Science, and have gained traciton in recent years. All 6 works (4.3%), using R were published from 2014 onwards, and studies using Python were published between 2019 and 2020. Therefore, we can conclude that Matlab is widely used and popular for MBSO projects. However, the recent and successful use of R and Python makes them good alternatives, mainly given the growth in communities focused on ML studies (Pedregosa et al., 2011), and newly created and improved algorithms, that may increase metamodel efficiency and fit.

The last item analyzed in this section was the performance metrics used to evaluate the metamodels, to answer RQ5 ("How are the results measured/validated?"). The three most widely used metrics were MSE (or root MSE), the coefficient of determination R, R^2, R^2_{adj}, MAPE, and MAE. More details on these metrics are given in Section 2.2.

The metamodel validation process is a fundamental step, since researchers can evaluate and compare the effectiveness of different metamodels based on simulated system representativeness. Of all the articles analyzed, 38.3% did not specify the type of validation process used, and we recommend that future research pay more attention to this stage, since validation is an important part of the MBSO methodological process, and provides support and reliability to metamodels. 22.9% of all articles (e.g., Sousa Junior et al., 2019; Hussain et al., 2015; Luo et al., 2019; Raei et al., 2019), used the database division strategy in two parts, the first for training the metamodel (around 70%-80% of the data), and the second for testing/validating (20%-30% of data).

However, cross-validation techniques were used in most studies, and 20 (16.4%), of all authors used leave-one-out cross-validation (e.g., Azizi et al., 2019; Xiqun et al., 2019), and 15 (12.3%), used k-fold cross-validation, (e.g., Coelho and Pinto, 2018; Hou et al., 2019; Wade, 2019). For a more detailed explanation of cross-validation techniques see Bergmeir and Benitez (2012).

4.3 MBSO Frameworks

Given the relevance and complexity of this topic, and the wide variety of joint techniques and applications for metamodeling, some MBSO frameworks have emerged to help researchers and practitioners with their projects. Existing frameworks, as proposed in literature, must be analyzed to propose a MBSO in SED framework to obtain insights that result in better assertiveness when designing suggested steps. Thus, the same keywords and search criteria/article screening used in this SLR were again applied to journal articles alone.

Osorio and Bierlaire (2013), proposed a framework optimizing urban traffic simulations, integrating metamodeling with quadratic polynomials and the derivative-free trust region optimization algorithm. Pedrielli et al. (2020), proposed a framework for real-time route optimization for ship collision detection systems. The method used metamodeling and agent-based simulations to generate contingency plans for possible collisions, minimizing risks and negative effects from route changes.

Other frameworks have also addressed specific engineering problems, e.g., Dang (2014), who used metamodeling, DOE, and Computer Aided Engineering to optimize plastic injection processes. Ranjbar et al. (2020), developed a multi-objective optimization procedure with metamodeling and NSGA II to manage water intrusion into coastal aquifers.

Parnianifard et al. (2020b), proposed a robust optimization framework to optimize OF while minimizing response variance using metamodeling, Taguchi orthogonal arrays, Pareto frontiers, and Bootstrapping sensitivity analysis.

Barton (2009) proposed a general optimization framework using metamodeling in SED models. The SO process was bifurcated into two possible strategies, as either local or global metamodeling, as shown in Figure 4.8. The global metamodeling strategy, which is the focus of this paper, adjusts the metamodel to represent the entire search space, and this global approximation is used in iterative optimization processes. The local strategy alternates the metamodeling and optimization steps, since as optimization moves in the search space, new local regions of θ space are explored, and new metamodel approximations are adjusted.

The global metamodeling strategy proposed by Barton (2009), was designed with seven main steps. In the first step, the search space region must be identified by defining the problem constraints. The type of metamodel is chosen in the second step. The author suggests using RNA, RBF, splines, or Kriging to do this. The following three steps deal with choosing the type of experiment - which can be performed via fractional or complete factorial arrangements, CCD, LHD, etc. - conducting the experiments in the simulation model, and adjusting the metamodel to the experimental arrangement. If the error metrics are unacceptable, one should go back to the third step and change the metamodel type or experimental design. The last two steps address the optimization phase, where the optimization algorithm must be applied, and the optimal simulation model response must be evaluated.

It is worth mentioning that there are several methodologies for metamodeling, however, there are two main identifiable aspects, i.e., fixed sampling and incremental sampling. In fixed sampling, a sample set with *N* experiments is generated at the beginning of the project and the metamodel is trained using this set. Figure 9 shows a fixed sampling metamodeling framework.

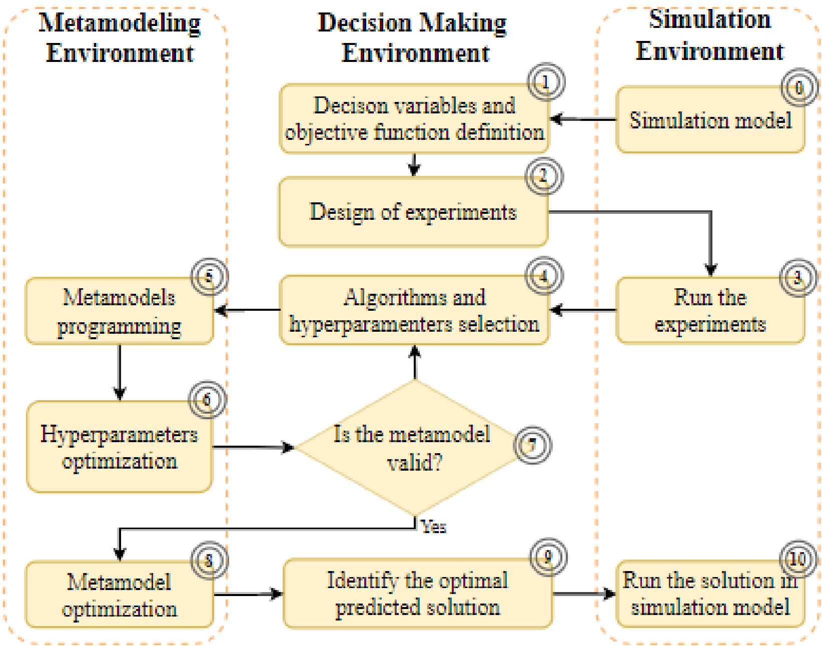

Figure 9: Fixed sampling metamodeling framework

The framework steps are as follows:

1. Determine the decision variables (xi, ..., xn), and their variation amplitudes, along with the OF that will be optimized (y);
2. Define the experimental arrangement and select a sample of m scenarios that will comprise the metamodel training base. The LHD array is usually suggested for this step;
3. Run experiments on the simulation model, and store the simulated data;
4. Select the ML algorithms and the hyper-parameters that will be optimized, along with the variation range;
5. Build the metamodel structure in the software or programming language;
6. Perform hyperparameter optimization by validating and testing using the cross-validation technique. The GA is usually suggested as an optimization method;
7. Evaluate whether the error and adjustment metrics are acceptable according to the decision-maker criteria. If yes, then go to step 8. If no, then go back to step 4;

8. Run the metamodel optimization to find the decision variable values that optimize the predicted OF value defined in Step 1. The decision-maker can use any optimization techniques they see fit, e.g., AG, Ranking and Selection, NSGA II, etc.;
9. Identify the solution that optimizes the OF predicted by the metamodel;
10. Evaluate the solution from the simulation model.

By contrast, the incremental sampling approach is adaptive, and performs experiments preferably in more promising solution space regions. A small sample set is used to train the metamodels, and these are used to find new promising trial points in the simulation. These new points are added to the first set, and the loop continues until meeting the stop criterion. Figure 10 shows the incremental sampling framework.

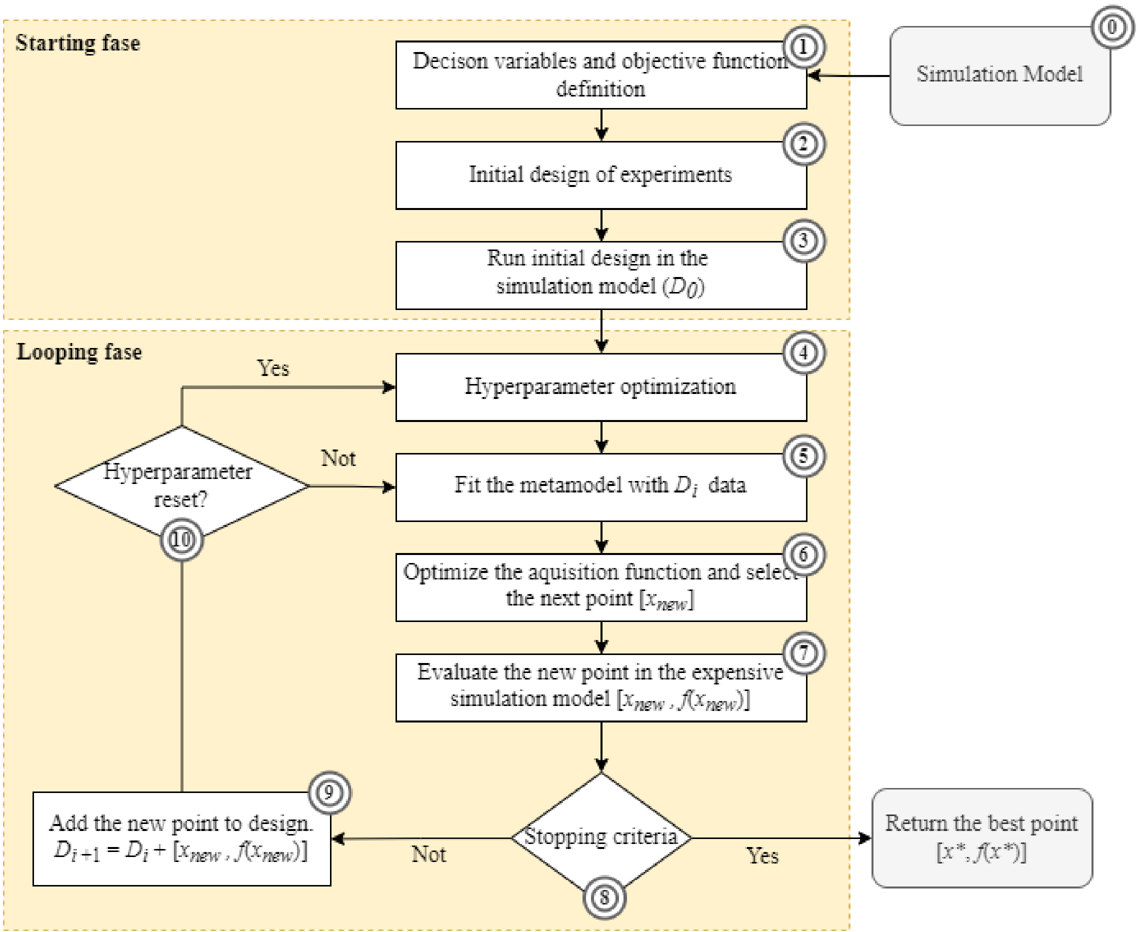

Figure 10: Incremental sampling metamodeling framework

The framework steps are as follows:

1) Determine the decision variables (*xi*, ..., *xn*), and their variation amplitudes, along with the OF that will be optimized (*y*);
2) Define the experimental setup and select a sample of *m* scenarios that will comprise the initial metamodels training base. The LHD array is usually suggested for this step;
3) Run experiments on the simulation model, and store the simulated data;

4) Optimize the hyper-parameters via validations and testing using the cross-validation technique on the selected metamodel and the initial generated experiment base. The GA is usually suggested as an optimization method;
5) Train the metamodel using available experiments;
6) Use the metamodel to predict the point that optimizes the acquisition function to find the point (or points) that should be tested in the simulation. The acquisition function can be used from simpler functions, e.g., the highest/smallest $\hat{f}(x)$ value, or more complex functions, e.g., Expected Improvement or Probability of improvement, which seek the point that best meets trade-offs between regions with higher global optimum ($\hat{f}(x)$high) probabilities, and regions with high uncertainty ($\hat{\sigma}(x)$high);
7) Evaluate this new point (or set of points) in the original simulation model;
8) Check if the stop criteria have been met. If yes, return the optimal point. If no, proceed to step 9. The maximum number of iterations or simulations, the metamodel reaching certain precision, or the maximum number of iterations without improving the optimal response, can all be used as stop criteria;
9) Add the new test point to the current test base;
10) Evaluate whether the metamodel's hyper-parameters need to be adjusted. Comparing the metamodel error of the current iteration with the previous one can be used as a criterion. If further hyper-parameter adjustments are needed, proceed to step 4. If not, proceed to step 5.

These frameworks were developed based on analyzing good practices observed in the articles from the SLR.

4.4 Future directions for MBSO research

Based on the articles studied in this SLR, we identified that trends and gaps in literature can constitute opportunities involving adopted MBSO contexts and methods. Regarding the methods used, growing development in the area of computational science has greatly contributed to improved learning algorithms, which can consequently be studied by creating more accurate metamodels. One algorithm that has been gaining traction in the ML field, given its flexibility and learning ability, is Deep Learning (Higham and Higham, 2019; Xu et al., 2020). However, it has not been used to create SO metamodels, meaning that future studies could explore this research opportunity.

Beyond Deep Learning, there are still many opportunities for using ML algorithms already addressed in literature, but which are still being developed. Classical algorithms like Kriging and ANNs have shown considerable growth in recent years, especially when varying how these algorithms are used, e.g., (Rojas Gonzalez, Jalali, and Van Nieuwenhuyse 2020; Yousefi et al., 2018), or by using them with other techniques to build innovative solutions, e.g., (Ranjbar et al., 2020; Yaohui, 2017). Extreme learning machines e.g., (Li et al., 2020), SVM (Li et al., 2019), and ensemble methods, e.g., (Lal and Datta 2020), are other algorithms that have gained recent attention in MBSO, have high learning capabilities, and are under active development in the field of artificial intelligence.

Hyper-parameter optimization techniques increase metamodel effectiveness in correctly representing simulation models. Optimization methods like AG, particle swarms, Bayesian optimizations, and search grids for hyper-parameter optimization have led to positive results in several studies, e.g. (Amiri et al., 2019; Hou et al., 2019; Hullen et al., 2020; Kim and Boukouvala, 2020; Li et al., 2019), but research is still incipient.

The global optimization method is another technique that has shown good results for building metamodels, which uses EGO and EI techniques to build adaptive and interactive metamodels. Kriging was mostly used as a metamodeling algorithm for applications involving this strategy, e.g., (Coelho and Pinto, 2018; Ivanova and Kuhnt, 2014; Wang et al., 2020). We recommend that future research explore this adaptive strategy for building metamodels based on other algorithms, like ANNs, SVM, Extreme Learning Machines, decision trees, etc.

Metamodeling has been used to address more complex and large-scale problems over the years. Feature selection techniques have been used more and more to deal with increasingly large databases in the ML and data mining communities. Feature selection can be interpreted as a set of techniques used to reduce problem size, removing irrelevant or redundant variables from the database (Bommert et al. 2020). However, these techniques were only addressed by Kim and Boukouvala (2020), and Sharif and Hammad (2019), and we suggest that future research address these techniques, like Principal Component Analysis, Wrappers, Filters, etc., to treat data in the MBSO process.

Regarding the context in which the studies were conducted, the highest growth trend was observed in the tertiary economic sector, driven mainly by metamodeling use in logistical problems, like cargo transportation and storage, and calibrating and optimizing of urban traffic models. However, it is worth mentioning that no MBSO application-optimized operations at distribution centers, a potential field for application, especially given e-commerce expansion and reverse logistics challenges. Furthermore, the COVID-19 pandemic intensified the pre-existing trends of using modeling and simulation techniques in healthcare (Currie et al., 2020), giving metamodeling an opportunity to reduce inherent healthcare model complexity and provide faster more consistent decision making.

Finally, with advancements in industry & service 4.0, many organizations from different sectors have demanded Digital Twin studies and applications. According to Mueller et al. (2017), Digital Twins are physical and virtual system junctions (simulators), connected in full or in partial autonomy, with real-time controls, which can work with human decision-making interfaces. This limits model-based SO technique use, since decision making is done in real time, or in short time intervals, leading to metamodeling replacing optimization simulation models via Digital Twin simulations. Dunke and Nickel (2020), Calahorrano et al. (2016), Pedrielli et al. (2020), Shirazi et al. (2011), and Steer et al. (2012), addressed operationally applying metamodels for decision making in near real time.

5 Final Considerations

This paper sought to (a) carry out an SLR on MBSO to (b) identify and explore the main aspects related to the topic, (c) analyze and summarize the results to answer the RQs, and (d) define the state-of-the arts, gaps, and trends that could be explored in future research.

SLR proved to be satisfactory in exploring and analyzing existing scientific literature. The guidelines proposed by Denyer and Tranfield (2009), and Denyer et al. (2008), greatly aided in better conducting this study, preventing possible distortions and author partialities in the results, thereby allowing us to meet our first objective. It is worth mentioning that this study did not seek to exhaustively analyze all articles on the subject, but to systematically analyze a representative sample that would allow us to consistently answer the RQs, generating insights into MBSO practices, to assist future researchers and practitioners in understanding and identifying the frontier of existing knowledge in the area.

According to our second and third objectives, Sections 4.1 and 4.2 presented the main aspects related to the MBSO contexts of the studies, along with the simulation, optimization, sampling, and metamodeling methods adopted in the studies. Finally, Section 4.3 addressed the main frameworks and best MBSO project practices. Section 4.4 discussed the gaps and trends identified in this study, thereby responding to our final proposed objective.

Regarding the RQs, we can start by concluding that metamodeling has reduced computational time spent on SO problems, where recursive models would be expensive, or unfeasible. Metamodeling has been applied to many problems, with emphasis on MBSO optimizations for process control parameters, sizing, resource allocation, urban traffic controls, and product designs.

MBSO studies were initiated predominantly by industry. However, they have tended to expand into primary sectors, especially for water extraction and treatment, and into the tertiary economic sector, focusing on smart city concepts and better healthcare management.

Our second conclusion deals with the adopted methods. We were unable to reach a global consensus on which method or technique is most suitable for each problem, since this decision is often made based on available resources and the prior knowledge of researchers. The most frequently used simulation techniques were SED, the finite element method, and microscopic simulations of urban traffic. Regarding the most frequently used optimization methods, Genetic Algorithms were used for dealing with single-objective problems, the NSGA II algorithm was used for multi-objective problems in SED models, the finite elements, finite volumes, and trust-region methods were used for traffic simulations.

Lastly, the most used algorithms for building metamodels were Kriging, PR, RNAs, RBF, and SVM, along with ensemble methods (Bagging, AdaBoost, etc.), which combine one or more metamodeling algorithm to formulate a metamodel that can learn better, e.g., RF and GBT. Also, LHD and Random design were the main sampling strategies for search spaces.

Acknowledgements

The authors are grateful to CAPES, CNPq, FAPEMIG, and the UNIFEI Postgraduate Program in Production Engineering for supporting this research.

References

Aghaei Chadegani, A., Salehi, H., Md Yunus, M. M., Farhadi, H., Fooladi, M., Farhadi, M., & Ale Ebrahim, N. (2013). A comparison between two main academic literature collections: Web of science and scopus databases. *Asian Social Science*, *9*(5), 18–26. https://doi.org/10.5539/ass.v9n5p18

Alabas, C., Altiparmak, F., & Dengiz, B. (2002). A comparison of the performance of artificial intelligence techniques for optimizing the number of kanbans. *Journal of the Operational Research Society*, *53*(8), 907–914. https://doi.org/10.1057/palgrave.jors.2601395

Albanesi, A., Roman, N., Bre, F., & Fachinotti, V. (2018). A metamodel-based optimization approach to reduce the weight of composite laminated wind turbine blades. *Composite Structures*, *194*(February), 345–356. https://doi.org/10.1016/j.compstruct.2018.04.015

Ali, W., Khan, M. S., Qyyum, M. A., & Lee, M. (2018). Surrogate-assisted modeling and optimization of a natural-gas liquefaction plant. *Computers and Chemical Engineering*, *118*, 132–142. https://doi.org/10.1016/j.compchemeng.2018.08.003

Alizadeh, M. R., Nikoo, M. R., & Rakhshandehroo, G. R. (2017). Developing a Multi-Objective Conflict-Resolution Model for Optimal Groundwater Management Based on Fallback Bargaining Models and Social Choice Rules: a Case Study. *Water Resources Management*, *31*(5), 1457–1472. https://doi.org/10.1007/s11269-017-1588-7

Amaral, J. V. S. do, de Carvalho Miranda, R., Montevechi, J. A. B., dos Santos, C. H., & Gabriel, G. T. (2022). Metamodeling-based simulation optimization in manufacturing problems: a comparative study. *The International Journal of Advanced Manufacturing Technology*, 20. https://doi.org/10.1007/s00170-022-09072-9

Amaran, S., Sahinidis, N. V., Sharda, B., & Bury, S. J. (2016). Simulation optimization: a review of algorithms and applications. *Annals of Operations Research*, *240*(1), 351–380. https://doi.org/10.1007/s10479-015-2019-x

Amiri, F., Shirazi, B., & Tajdin, A. (2019). Multi-objective simulation optimization for uncertain resource assignment and job sequence in automated flexible job shop. *Applied Soft Computing Journal*, *75*, 190–202. https://doi.org/10.1016/j.asoc.2018.11.015

Aouadni, S., Aouadni, I., & Rebaï, A. (2019). A systematic review on supplier selection and order allocation problems. *Journal of Industrial Engineering International*, *15*(2009), 267–289. https://doi.org/10.1007/s40092-019-00334-y

Ataie-Ashtiani, B., Ketabchi, H., & Rajabi, M. M. (2014). Optimal Management of a Freshwater Lens in a Small Island Using Surrogate Models and Evolutionary Algorithms. *Journal of Hydrologic Engineering*, *19*(2), 339–354. https://doi.org/10.1061/(asce)he.1943-5584.0000809

Aydin, C. S., Ozgurler, S., Durmusoglu, M. B., & Ozgurler, M. (2018). Response surface approach to robust design of assembly cells through simulation. *Assembly Automation*, *38*(4), 450–464. https://doi.org/10.1108/AA-08-2017-093

Azizi, M. J., Seifi, F., & Moghadam, S. (2019). A robust simulation optimization algorithm using kriging and particle swarm optimization: Application to surgery room optimization. *Communications in Statistics: Simulation and Computation*, *0*(0), 1–17. https://doi.org/10.1080/03610918.2019.1593452

Bălan, C. (2018). The disruptive impact of future advanced ICTs on maritime transport: a systematic review. *Supply Chain Management*, *25*(2), 157–175. https://doi.org/10.1108/SCM-03-2018-0133

Bananmah, M., Reza Nikoo, M., Nematollahi, B., & Sadegh, M. (2020). Optimizing chute-flip bucket system based on meta-modelling approach. *Canadian Journal of Civil Engineering*, *47*(5), 584–595. https://doi.org/10.1139/cjce-2018-0534

Baquela, E. G., & Olivera, A. C. (2019). A novel hybrid multi-objective metamodel-based evolutionary optimization algorithm. *Operations Research Perspectives*, *6*(November 2018), 100098. https://doi.org/10.1016/j.orp.2019.100098

Barton, R. R. (2009a). Simulation Optimization Using Metamodel. *Winter Simulation Conference*, *2*, 230–238.

Barton, R. R. (2009b). Simulation Optimization Using Metamodel. In *Winter Simulation Conference* (Issue 2).

Barton, R. R., & Meckesheimer, M. (2006). Chapter 18 Metamodel-Based Simulation Optimization. In *Handbooks in Operations Research and Management Science* (Vol. 13, Issue C, pp. 535–574). https://doi.org/10.1016/S0927-0507(06)13018-2

Bartz-Beielstein, T., Zaefferer, M., & Pham, Q. C. (2018). Optimization via multimodel simulation: A new approach to optimization of cyclone separator geometries. *Structural and Multidisciplinary Optimization*, *58*(3), 919–933. https://doi.org/10.1007/s00158-018-1934-2

Bashiri, M., & Geranmayeh, A. F. (2011). Tuning the parameters of an artificial neural network using central composite design and genetic algorithm. *Scientia Iranica*, *18*(6), 1600–1608. https://doi.org/10.1016/j.scient.2011.08.031

Belgin, O. (2019). Data Envelopment Analysis Based Metamodeling for Multi Objective Simulation Optimization in a Manufacturing Line. *Sigma Journal of Engineering and Natural Sciences*, *37*(4), 1435–1449.

Bergmeir, C., & Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, *191*, 192–213. https://doi.org/10.1016/j.ins.2011.12.028

Bin Makhashen, Y., Rafi-ul-Shan, P. M., Bashiri, M., Hasan, R., Amar, H., & Khan, M. N. (2020). Exploring the role of ambidexterity and coopetition in designing resilient fashion supply chains: a multi-evidence-based approach. *Journal of Enterprise Information Management*. https://doi.org/10.1108/JEIM-08-2019-0213

Bommert, A., Sun, X., Bischl, B., Rahnenführer, J., & Lang, M. (2020). Benchmark for filter methods for feature selection in high-dimensional classification data. *Computational Statistics and Data Analysis*, *143*, 106839. https://doi.org/10.1016/j.csda.2019.106839

Booth, A. (2006). Clear and present questions: formulating questions for evidence based practice. *Library Hi Tech*, *24*(3), 355–368. https://doi.org/10.1108/07378831111138242

Boukouvala, F., & Ierapetritou, M. G. (2013). Surrogate-based optimization of expensive flowsheet modeling for continuous pharmaceutical manufacturing. *Journal of Pharmaceutical Innovation*, *8*(2), 131–145. https://doi.org/10.1007/s12247-013-9154-1

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32. https://doi.org/10.1201/9780429469275-8

Cai, X., Qiu, H., Gao, L., Jiang, C., & Shao, X. (2019). An efficient surrogate-assisted particle swarm optimization algorithm for high-dimensional expensive problems. *Knowledge-Based Systems*, *184*, 104901. https://doi.org/10.1016/j.knosys.2019.104901

Candelieri, A., Perego, R., & Archetti, F. (2018). Bayesian optimization of pump operations in water distribution systems. *Journal of Global Optimization*, *71*(1), 213–235. https://doi.org/10.1007/s10898-018-0641-2

Cao, Y., Xu, Y., & Du, J. (2020). Multi-variable estimation-based safe screening rule for small sphere and large margin support vector machine. *Knowledge-Based Systems*, *191*(xxxx), 105223. https://doi.org/10.1016/j.knosys.2019.105223

Carson, Y., & Maria, A. (1997). Simulation optimization: Methods and applications. *Winter Simulation Conference Proceedings*, 118–126.

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, *7*(3), 1247–1250. https://doi.org/10.5194/gmd-7-1247-2014

Chang, K. H., & Hsieh, L. Y. (2016). Determination of Wafer Start Mix in Semiconductor Manufacturing during New Technology Ramp-Up: Model, Solution Method, and an Empirical Study. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *46*(2), 294–302. https://doi.org/10.1109/TSMC.2015.2426174

Chen, M., Izady, A., & Abdalla, O. A. (2017). An efficient surrogate-based simulation-optimization method for calibrating a regional MODFLOW model. *Journal of Hydrology*, *544*, 591–603. https://doi.org/10.1016/j.jhydrol.2016.12.011

Chen, X., He, X., Xiong, C., Zhu, Z., & Zhang, L. (2019). A Bayesian stochastic kriging optimization model dealing with heteroscedastic simulation noise for freeway traffic management. *Transportation Science*, *53*(2), 545–565. https://doi.org/10.1287/trsc.2018.0819

Chen, X. M., Xiong, C., He, X., Zhu, Z., & Zhang, L. (2016). Time-of-day vehicle mileage fees for congestion mitigation and revenue generation: A simulation-based optimization method and its real-world application. *Transportation Research Part C: Emerging Technologies*, *63*, 71–95. https://doi.org/10.1016/j.trc.2015.12.001

Chen, X., Osorio, C., & Santos, B. F. (2019). Simulation-based travel time reliable signal control. *Transportation Science*, *53*(2), 523–544. https://doi.org/10.1287/trsc.2017.0812

Chen, X., Yin, M., Song, M., Zhang, L., & Li, M. (2014). Social welfare maximization of multimodal transportation: Theory, metamodel, and application to Tianjin Ecocity, China. *Transportation Research Record*, *2451*(2451), 36–49. https://doi.org/10.3141/2451-05

Cherkassky, V., & Ma, Y. (2004). Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks*, *17*(1), 113–126. https://doi.org/10.1016/S0893-6080(03)00169-2

Chong, L., & Osorio, C. (2018). A simulation-based optimization algorithm for dynamic large-scale urban transportation problems. *Transportation Science*, *52*(3), 637–656. https://doi.org/10.1287/trsc.2016.0717

Christelis, V., & Mantoglou, A. (2016). Pumping Optimization of Coastal Aquifers Assisted by Adaptive Metamodelling Methods and Radial Basis Functions. *Water Resources Management*, *30*(15), 5845–5859. https://doi.org/10.1007/s11269-016-1337-3

Ciuffo, B., & Punzo, V. (2014). “No free lunch” Theorems applied to the calibration of traffic simulation models. *IEEE Transactions on Intelligent Transportation Systems*, *15*(2), 553–562. https://doi.org/10.1109/TITS.2013.2287720

Coelho, G. F., & Pinto, L. R. (2018). Kriging-based simulation optimization: An emergency medical system application. *Journal of the Operational Research Society*, *69*(12), 2006–2020. https://doi.org/10.1080/01605682.2017.1418149

Costa, E., Soares, A. L., & de Sousa, J. P. (2020). Industrial business associations improving the internationalisation of SMEs with digital platforms: A design science research approach. *International Journal of Information Management*, *53*(January), 102070. https://doi.org/10.1016/j.ijinfomgt.2020.102070

Crisci, C., Ghattas, B., & Perera, G. (2012). A review of supervised machine learning algorithms and their applications to ecological data. *Ecological Modelling*, *240*, 113–122. https://doi.org/10.1016/j.ecolmodel.2012.03.001

Cuckler, R., Chang, K. H., & Hsieh, L. Y. (2017). Optimal Parallel Machine Allocation Problem in IC Packaging Using IC-PSO: An Empirical Study. *Asia-Pacific Journal of Operational Research*, *34*(6), 1–20. https://doi.org/10.1142/S0217595917500348

Currie, C. S. M., Fowler, J. W., Kotiadis, K., Monks, T., Onggo, B. S., Robertson, D. A., & Tako, A. A. (2020). How simulation modelling can help reduce the impact of COVID-19. *Journal of Simulation*, *14*(2), 83–97. https://doi.org/10.1080/17477778.2020.1751570

Dai, B., Gu, C., Zhao, E., & Qin, X. (2018). Statistical model optimized random forest regression model for concrete dam deformation monitoring. *Structural Control and Health Monitoring*, *25*(6), 2170. https://doi.org/10.1002/stc.2170

Dang, X. P. (2014). General frameworks for optimization of plastic injection molding process parameters. *Simulation Modelling Practice and Theory*, *41*, 15–27. https://doi.org/10.1016/j.simpat.2013.11.003

De Jong, K. A. (1975). *Analysis of the Behavior of a Class of Genetic Adaptive Systems*.

De La Fuente, R., Erazo, I., & Smith, R. L. (2019). Enabling intelligent processes in simulation utilizing the tensorflow deep learning resources. *Proceedings - Winter Simulation Conference*, *2018-Decem*, 1108–1119. https://doi.org/10.1109/WSC.2018.8632539

De La Fuente, R., & Smith, R. (2017). Metamodeling a system dynamics model: a contemporany comparison of methods. *Winter Simulation ConferenceWinter Simulation Conference*, *53*(9), 1689–1699. https://doi.org/10.1017/CBO9781107415324.004

Dellino, G., Kleijnen, J. P. C., & Meloni, C. (2015). Metamodel-based robust simulation-optimization: An overview. *Operations Research/ Computer Science Interfaces Series*, *59*, 27–54. https://doi.org/10.1007/978-1-4899-7547-8_2

Dengiz, B., Alabas-Uslu, C., & Dengiz, O. (2009). Optimization of manufacturing systems using a neural network metamodel with a new training approach. *Journal of the Operational Research Society*, *60*(9), 1191–1197. https://doi.org/10.1057/palgrave.jors.2602620

Dengiz, B., Bektas, T., & Ultanir, A. E. (2006). Simulation optimization based DSS application: A diamond tool production line in industry. *Simulation Modelling Practice and Theory*, *14*(3), 296–312. https://doi.org/10.1016/j.simpat.2005.07.001

Dengiz, B., Iç, Y. T., & Belgin, O. (2015). A meta-model based simulation optimization using hybrid simulation-analytical modeling to increase the productivity in automotive industry. *Mathematics and Computers in Simulation*, *120*, 120–128. https://doi.org/10.1016/j.matcom.2015.07.005

Denyer, D., & Tranfield, D. (2009). Producing a systematic review. In *In D. A. Buchanan & A. Bryman (Eds.), The Sage handbook of organizational research methods* (pp. 671–689).

Denyer, D., Tranfield, D., & Van Aken, J. E. (2008). Developing design propositions through research synthesis. *Organization Studies*, *29*(3), 393–413. https://doi.org/10.1177/0170840607088020

Dhar, A., & Datta, B. (2009). Saltwater Intrusion Management of Coastal Aquifers. I: Linked Simulation-Optimization. *Journal of Hydrologic Engineering*, *14*(12), 1263–1272. https://doi.org/10.1061/(asce)he.1943-5584.0000097

Dunke, F., & Nickel, S. (2020). Neural networks for the metamodeling of simulation models with online decision making. *Simulation Modelling Practice and Theory*, *99*(September 2019), 102016. https://doi.org/10.1016/j.simpat.2019.102016

Figueira, G., & Almada-Lobo, B. (2014). Hybrid simulation-optimization methods: A taxonomy and discussion. *Simulation Modelling Practice and Theory*, *46*, 118–134. https://doi.org/10.1016/j.simpat.2014.03.007

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics and Data Analysis*, *38*(4), 367–378. https://doi.org/10.1016/S0167-9473(01)00065-2

Fu, M. C. (2002). Feature Article: Optimization for simulation: Theory vs. Practice. *INFORMS Journal on Computing*, *14*(3), 192–215. https://doi.org/10.1287/ijoc.14.3.192.113

Fu, Y., & Sahin, K. H. (2004). Better Optimization of Nonlinear Uncertain Systems (BONUS) for vehicle structural design. *Annals of Operations Research*, *132*(1–4), 69–84. https://doi.org/10.1023/B:ANOR.0000045277.22566.5f

Ganjisaffar, Y., Caruana, R., & Lopes, C. V. (2011). Bagging gradient-boosted trees for high precision, low variance ranking models. *SIGIR'11 - Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, *c*, 85–94. https://doi.org/10.1145/2009916.2009932

Ghorbani Mooselu, M., Nikoo, M. R., Bakhtiari Rayani, N., & Izady, A. (2019). Fuzzy Multi-Objective Simulation-Optimization of Stepped Spillways Considering Flood Uncertainty. *Water Resources Management*, *33*(7), 2261–2275. https://doi.org/10.1007/s11269-019-02263-2

Greenwood, A. G., Rees, L. P., & Siochi, F. C. (1998). An investigation of the behavior of simulation response surfaces. *European Journal of Operational Research*, *110*(2), 282–313. https://doi.org/10.1016/S0377-2217(97)00255-5

Greinacher, S., Overbeck, L., Kuhnle, A., Krahe, C., & Lanza, G. (2020). Multi-objective optimization of lean and resource efficient manufacturing systems. *Production Engineering*, *14*(2), 165–176. https://doi.org/10.1007/s11740-019-00945-9

Guo, R., & Zhang, Y. (2014). Exploration of correlation between environmental factors and mobility at signalized intersections. *Transportation Research Part D: Transport and Environment*, *32*, 24–34. https://doi.org/10.1016/j.trd.2014.05.011

Hachicha, W. (2011). A simulation metamodelling based neural networks for lot-sizing problem in MTO sector. *International Journal of Simulation Modelling*, *10*(4), 191–203. https://doi.org/10.2507/IJSIMM10(4)3.188

Hannah, L. A., Powell, W. B., & Ducson, D. B. (2014). SEMICONVEX REGRESSION FOR METAMODELING-BASED OPTIMIZATION∗. *SIAM J. OPTIM.*, *24*(2), 573–597.

Hassannayebi, E., Boroun, M., Alaei Jordehi, S., & Kor, H. (2019). Train schedule optimization in a high-speed railway system using a hybrid simulation and meta-model approach. *Computers and Industrial Engineering*, *138*(July), 106110. https://doi.org/10.1016/j.cie.2019.106110

Hassoun, M. H. (1995). Fundamentals of Artificial Neural Networks. In *MIT Press*. https://doi.org/10.1177/0261018311403863

Hastie, T., Tibshirani, R., & Friedman, J. (2008). The Elements of Statistical Learnning. In *Springer* (Vol. 2).

Helleno, A. L., Pimentel, C. A., Ferro, R., Santos, P. F., Oliveira, M. C., & Simon, A. T. (2015). Integrating value stream mapping and discrete events simulation as decision making tools in operation management. *International Journal of Advanced Manufacturing Technology, 80*(5–8), 1059–1066. https://doi.org/10.1007/s00170-015-7087-1

Higham, C. F., & Higham, D. J. (2019). Deep learning: An introduction for applied mathematicians. *SIAM Review, 61*(4), 860–891. https://doi.org/10.1137/18M1165748

Hoad, K., Monks, T., & O'Brien, F. (2015). The use of search experimentation in discrete-event simulation practice. *Journal of the Operational Research Society, 66*(7), 1155–1168. https://doi.org/10.1057/jors.2014.79

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence* (1º). MIT Press.

Hong, J. H., Seo, K. M., & Kim, T. G. (2013). Simulation-based optimization for design parameter exploration in hybrid system: A defense system example. *Simulation*, *89*(3), 362–380. https://doi.org/10.1177/0037549712466707

Horng, S. C., & Lin, S. S. (2017). Ordinal optimization based metaheuristic algorithm for optimal inventory policy of assemble-to-order systems. *Applied Mathematical Modelling*, *42*, 43–57. https://doi.org/10.1016/j.apm.2016.10.002

Horng, S. C., & Lin, S. S. (2018). Embedding Advanced Harmony Search in Ordinal Optimization to Maximize Throughput Rate of Flow Line. *Arabian Journal for Science and Engineering*, *43*(2), 1015–1031. https://doi.org/10.1007/s13369-017-2864-9

Hou, Z., Dai, Z., Lao, W., Wang, Y., & Lu, W. (2019). Application of Mixed-Integer Nonlinear Optimization Programming Based on Ensemble Surrogate Model for Dense Nonaqueous Phase Liquid Source Identification in Groundwater. *Environmental Engineering Science*, *36*(6), 699–709. https://doi.org/10.1089/ees.2018.0366

Hu, B., Palta, M., & Shao, J. (2006). Properties of R2 statistics for logistic regression. *Statistics in Medicine*, *25*(8), 1383–1395. https://doi.org/10.1002/sim.2300

Hüllen, G., Zhai, J., Kim, S. H., Sinha, A., Realff, M. J., & Boukouvala, F. (2020). Managing uncertainty in data-driven simulation-based optimization. *Computers and Chemical Engineering*, *136*, 106519. https://doi.org/10.1016/j.compchemeng.2019.106519

Hurrion, R. D. (1997). An example of simulation optimisation using a neural network metamodel: Finding the optimum number of kanbans in a manufacturing system. *Journal of the Operational Research Society*, *48*(11), 1105–1112. https://doi.org/10.1057/palgrave.jors.2600468

Hussain, M. S., Javadi, A. A., Ahangar-Asr, A., & Farmani, R. (2015). A surrogate model for simulation-optimization of aquifer systems subjected to seawater intrusion. *Journal of Hydrology*, *523*, 542–554. https://doi.org/10.1016/j.jhydrol.2015.01.079

Huyet, A. L., & Paris, J. L. (2001). Optimizing simulated manufacturing systems using machine learning coupled to evolutionary algorithms. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, *1*, 17–21. https://doi.org/10.1109/ETFA.2001.996349

Ilk, N., Shang, G., & Goes, P. (2020). Improving customer routing in contact centers: An automated triage design based on text analytics. *Journal of Operations Management*, *66*(5), 553–577. https://doi.org/10.1002/joom.1084

Ivanova, M., & Kuhnt, S. (2014). A parallel optimization algorithm based on FANOVA decomposition. *Quality and Reliability Engineering International*, *30*(7), 961–974. https://doi.org/10.1002/qre.1710

Jackson, I., Tolujevs, J., Lang, S., & Kegenbekov, Z. (2019). Metamodelling of Inventory-Control Simulations Based on a Multilayer Perceptron. *Transport and Telecommunication*, *20*(3), 251–259. https://doi.org/10.2478/ttj-2019-0021

Jagannatham, A. K., & Kumar, V. (2016). Introduction to game theory. In *Decision Sciences: Theory and Practice* (pp. 79–144). https://doi.org/10.1201/9781315183176

Jalali, H., & Nieuwenhuyse, I. Van. (2015). Simulation optimization in inventory replenishment: A classification. *IIE Transactions*, *47*(11), 1217–1235. https://doi.org/10.1080/0740817X.2015.1019162

Jeng, J. T. (2006). Hybrid approach of selecting hyperparameters of support vector machine for regression. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *36*(3), 699–709. https://doi.org/10.1109/TSMCB.2005.861067

Kim, S. H., & Boukouvala, F. (2020). Machine learning-based surrogate modeling for data-driven optimization: a comparison of subset selection for regression techniques. *Optimization Letters*, *14*(4), 989–1010. https://doi.org/10.1007/s11590-019-01428-7

Kim, W., Jeon, Y., & Kim, Y. (2016). Simulation-based optimization of an integrated daylighting and HVAC system using the design of experiments method. *Applied Energy*, *162*, 666–674. https://doi.org/10.1016/j.apenergy.2015.10.153

Kleijnen, J. P. C. (2014). Simulation-optimization via Kriging and bootstrapping: a survey. *Journal of Simulation*, *8*(4), 241–250. https://doi.org/10.1057/jos.2014.4

Kleijnen, J. P. C. (2017a). Regression and Kriging metamodels with their experimental designs in simulation: A review. *European Journal of Operational Research*, *256*(1), 1–16. https://doi.org/10.1016/j.ejor.2016.06.041

Kleijnen, J. P. C. (2017b). Simulation Optimization Through Regression or Kriging Metamodels. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2969730

Kloda, L. A., Boruff, J. T., & Cavalcante, A. S. (2020). A comparison of patient, intervention, comparison, outcome (PICO) to a new, alternative clinical question framework for search skills, search results, and self-efficacy: A randomized controlled trial. *Journal of the Medical Library Association*, *108*(2), 185–194. https://doi.org/10.5195/jmla.2020.739

Kochan, C. G., & Nowicki, D. R. (2018). Supply chain resilience: a systematic literature review and typological framework. *International Journal of Physical Distribution and Logistics Management*, *48*(8), 842–865. https://doi.org/10.1108/IJPDLM-02-2017-0099

Krige, D. G. (1951). A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand. *Journal of the Chemical Metallurgical & Society of South Mining Africa*, *52*(6), 119–139.

Krityakierne, T., Akhtar, T., & Shoemaker, C. A. (2016). SOP: parallel surrogate global optimization with Pareto center selection for computationally expensive single objective problems. *Journal of Global Optimization*, *66*(3), 417–437. https://doi.org/10.1007/s10898-016-0407-7

Lal, A., & Datta, B. (2020). Performance Evaluation of Homogeneous and Heterogeneous Ensemble Models for Groundwater Salinity Predictions: a Regional-Scale Comparison Study. *Water, Air, and Soil Pollution*, *231*(6). https://doi.org/10.1007/s11270-020-04693-w

Law, A. M. (2013). Simulation Modeling and Analysis, FIFTH EDITION. In *Simulation Modeling and Analysis*.

Lessmann, S., Stahlbock, R., & Crone, S. F. (2006). Genetic algorithms for support vector machine model selection. *IEEE International Conference on Neural Networks - Conference Proceedings*, 3063–3069. https://doi.org/10.1109/ijcnn.2006.247266

Levy, Y., & Ellis, T. J. (2006). A systems approach to conduct an effective literature review in support of information systems research. *Informing Science*, *9*(May 2014), 181–211. https://doi.org/10.28945/479

Li, G., Hwai-yong Tan, M., & Hui Ng, S. (2019). Metamodel-based optimization of stochastic computer models for engineering design under uncertain objective function. *IISE Transactions*, *51*(5), 517–530. https://doi.org/10.1080/24725854.2018.1504355

Li, J., Lu, W., Wang, H., Fan, Y., & Chang, Z. (2020). Groundwater contamination source identification based on a hybrid particle swarm optimization-extreme learning machine. *Journal of Hydrology*, *584*, 124657. https://doi.org/10.1016/j.jhydrol.2020.124657

Li, W., Wolinski, D., & Lin, M. C. (2017). City-scale traffic animation using statistical learning and metamodel-based optimization. *ACM Transactions on Graphics*, *36*(6), 1–12. https://doi.org/10.1145/3130800.3130847

Li, Y. F., Ng, S. H., Xie, M., & Goh, T. N. (2010). A systematic comparison of metamodeling techniques for simulation optimization in Decision Support Systems.

Applied Soft Computing Journal, *10*(4), 1257–1273. https://doi.org/10.1016/j.asoc.2009.11.034

Li, Y. S., Chi, H., Shao, X. Y., Qi, M. L., & Xu, B. G. (2020). A novel random forest approach for imbalance problem in crime linkage. *Knowledge-Based Systems*, *195*(xxxx), 105738. https://doi.org/10.1016/j.knosys.2020.105738

Lin, J. T., & Chen, C. M. (2015). Simulation optimization approach for hybrid flow shop scheduling problem in semiconductor back-end manufacturing. *Simulation Modelling Practice and Theory*, *51*, 100–114. https://doi.org/10.1016/j.simpat.2014.10.008

Luo, J., Ji, Y., & Lu, W. (2019). Comparison of Surrogate Models Based on Different Sampling Methods for Groundwater Remediation. *Journal of Water Resources Planning and Management*, *145*(5), 04019015. https://doi.org/10.1061/(asce)wr.1943-5452.0001062

Lupera Calahorrano, G. J., Shokry, A., Campanya, G., & Espuña, A. (2016). Application of the Meta-Multiparametric methodology to the control of emissions in the industry under continuous and discrete uncertain parameters. *Chemical Engineering Research and Design*, *115*, 365–373. https://doi.org/10.1016/j.cherd.2016.09.006

Lv, Y., Zhang, X., Kang, W., & Duan, Y. (2015). Managing Emergency Traffic Evacuation with a Partially Random Destination Allocation Strategy: A Computational-Experiment-Based Optimization Approach. *IEEE Transactions on Intelligent Transportation Systems*, *16*(4), 2182–2191. https://doi.org/10.1109/TITS.2015.2399852

McKay, M. D., Beckman, R. J., & Conover, W. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, *21*(2), 239–245. https://doi.org/https://doi.org/10.1080/00401706.1979.10489755

Methley, A. M., Campbell, S., Chew-Graham, C., McNally, R., & Cheraghi-Sohi, S. (2014). PICO, PICOS and SPIDER: A comparison study of specificity and sensitivity in three search tools for qualitative systematic reviews. *BMC Health Services Research*, *14*(1). https://doi.org/10.1186/s12913-014-0579-0

Miranda, R. D. C., Montevechi, J. A. B., Silva, A. F. Da, & Marins, F. A. S. (2014). A new approach to reducing search space and increasing efficiency in simulation optimization problems via the fuzzy-DEA-BCC. *Mathematical Problems in Engineering*, *2014*. https://doi.org/10.1155/2014/450367

Miranda, R. de C., Montevechi, J. A. B., da Silva, A. F., & Marins, F. A. S. (2017). Increasing the efficiency in integer simulation optimization: Reducing the search space through data envelopment analysis and orthogonal arrays. *European Journal of Operational Research*, *262*(2), 673–681. https://doi.org/10.1016/j.ejor.2017.04.016

Mirfenderesgi, G., & Mousavi, S. J. (2016). Adaptive meta-modeling-based simulation optimization in basin-scale optimum water allocation: A comparative analysis of meta-models. *Journal of Hydroinformatics*, *18*(3), 446–465. https://doi.org/10.2166/hydro.2015.157

Moghaddam, S., & Mahlooji, H. (2016). Robust simulation optimization using φ-divergence. *International Journal of Industrial Engineering Computations*, *7*(4), 517–534. https://doi.org/10.5267/j.ijiec.2016.5.003

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. https://doi.org/10.1080/03610918.2015.1134567

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. https://doi.org/10.1057/jors.2013.73

Mongeon, P., & Paul-Hus, A. (2016). The journal coverage of Web of Science and Scopus: a comparative analysis. *Scientometrics*, *106*(1), 213–228. https://doi.org/10.1007/s11192-015-1765-5

Monostori, L., & Viharos, Z. J. (2001). Hybrid, al- and simulation-supported optimisation of process chains and production plants. *CIRP Annals - Manufacturing Technology*, *50*(1), 353–356. https://doi.org/10.1016/S0007-8506(07)62138-6

Mueller, E., Chen, X. L., & Riedel, R. (2017). Challenges and Requirements for the Application of Industry 4.0: A Special Insight with the Usage of Cyber-Physical System. *Chinese Journal of Mechanical Engineering (English Edition)*, *30*(5), 1050–1057. https://doi.org/10.1007/s10033-017-0164-7

Myers, R. H., Montgomery, D. C., & Anderson-Cook, C. M. (2016). *Response surface methodology : process and product optimization using designed experiments* (4º). John Wiley & Sons.

Nguyen-Tuong, D., Seeger, M., & Peters, J. (2009). Model learning with local Gaussian process regression. *Advanced Robotics*, *23*(15), 2015–2034. https://doi.org/10.1163/016918609X12529286896877

Oliveira, A. L. I., Braga, P. L., Lima, R. M. F., & Cornélio, M. L. (2010). GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation. *Information and Software Technology*, *52*(11), 1155–1166. https://doi.org/10.1016/j.infsof.2010.05.009

Oliveira, J. B., Lima, R. S., & Montevechi, J. A. B. (2016). Perspectives and relationships in Supply Chain Simulation: A systematic literature review. *Simulation Modelling Practice and Theory*, *62*, 166–191. https://doi.org/10.1016/j.simpat.2016.02.001

Osorio, C. (2019a). Dynamic origin-destination matrix calibration for large-scale network simulators. *Transportation Research Part C: Emerging Technologies*, *98*(September 2018), 186–206. https://doi.org/10.1016/j.trc.2018.09.023

Osorio, C. (2019b). High-dimensional offline origin-destination (OD) demand calibration for stochastic traffic simulators of large-scale road networks. *Transportation Research Part B: Methodological*, *124*, 18–43. https://doi.org/10.1016/j.trb.2019.01.005

Osorio, C., & Bierlaire, M. (2013). A simulation-based optimization framework for urban transportation problems. *Operations Research*, *61*(6), 1333–1345. https://doi.org/10.1287/opre.2013.1226

Osorio, C., & Chong, L. (2015). A computationally efficient simulation-based optimization algorithm for large-scale urban transportation problems. *Transportation Science*, *49*(3), 623–636. https://doi.org/10.1287/trsc.2014.0550

Osorio, C., & Nanduri, K. (2015). Urban transportation emissions mitigation: Coupling high-resolution vehicular emissions and traffic models for traffic signal optimization. *Transportation Research Part B: Methodological*, *81*, 520–538. https://doi.org/10.1016/j.trb.2014.12.007

Osorio, C., & Punzo, V. (2019). Efficient calibration of microscopic car-following models for large-scale stochastic network simulators. *Transportation Research Part B: Methodological*, *119*, 156–173. https://doi.org/10.1016/j.trb.2018.09.005

Østergård, T., Jensen, R. L., & Maagaard, S. E. (2018). A comparison of six metamodeling techniques applied to building performance simulations. *Applied Energy*, *211*(October 2017), 89–103. https://doi.org/10.1016/j.apenergy.2017.10.102

Ouyang, Q., Lu, W., Lin, J., Deng, W., & Cheng, W. (2017). Conservative strategy-based ensemble surrogate model for optimal groundwater remediation design at DNAPLs-contaminated sites. *Journal of Contaminant Hydrology*, *203*(May), 1–8. https://doi.org/10.1016/j.jconhyd.2017.05.007

Parnianifard, A., & Azfanizam, A. S. (2020). Metamodel-based robust simulation-optimization assisted optimal design of multiloop integer and fractional-order PID controller. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, *33*(1), 1–20. https://doi.org/10.1002/jnm.2679

Parnianifard, A., Azfanizam, A. S., Ariffin, M. K. A., & Ismail, M. I. S. (2020a). Comparative study of metamodeling and sampling design for expensive and semi-expensive simulation models under uncertainty. *Simulation*, *96*(1), 89–110. https://doi.org/10.1177/0037549719846988

Parnianifard, A., Azfanizam, A. S., Ariffin, M. K. A., & Ismail, M. I. S. (2020b). Crossing weighted uncertainty scenarios assisted distribution-free metamodel-based robust simulation optimization. *Engineering with Computers*, *36*(1), 139–150. https://doi.org/10.1007/s00366-018-00690-0

Parnianifard, A., Azfanizam, A. S., Ariffin, M. K. A., Ismail, M. I. S., & Ale Ebrahim, N. (2019). Recent developments in metamodel based robust black-box simulation optimization: An overview. *Decision Science Letters*, *8*(1), 17–44. https://doi.org/10.5267/j.dsl.2018.5.004

Parnianifard, A., Azfanizam, A. S., Ariffin, M. K. A., Ismail, M. I. S., Maghami, M. R., & Gomes, C. (2018). Kriging and Latin hypercube sampling assisted simulation optimization in optimal design of PID controller for speed control of DC motor. *Journal* *of Computational and Theoretical Nanoscience*, *15*(5), 1471–1479. https://doi.org/10.1166/jctn.2018.7379

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., & Cournapeau, D. (2011). Scikit-learn: Machine Learning in Python. *Journal OfMachine Learning Research*, *12*(1), 2825–2830. https://doi.org/10.1145/2786984.2786995

Pedrielli, G., Xing, Y., Peh, J. H., Koh, K. W., & Ng, S. H. (2020). A real time simulation optimization framework for vessel collision avoidance and the case of singapore strait. *IEEE Transactions on Intelligent Transportation Systems*, *21*(3), 1204–1215. https://doi.org/10.1109/TITS.2019.2903824

Quan, N., Yin, J., Ng, S. H., & Lee, L. H. (2013). Simulation optimization via kriging: A sequential search using expected improvement with computing budget constraints. *IIE Transactions (Institute of Industrial Engineers)*, *45*(7), 763–780. https://doi.org/10.1080/0740817X.2012.706377

Raei, E., Reza Alizadeh, M., Reza Nikoo, M., & Adamowski, J. (2019). Multi-objective decision-making for green infrastructure planning (LID-BMPs) in urban storm water management under uncertainty. *Journal of Hydrology*, *579*(January), 124091. https://doi.org/10.1016/j.jhydrol.2019.124091

Ranjbar, A., Mahjouri, N., & Cherubini, C. (2020). Development of an efficient conjunctive meta-model-based decision-making framework for saltwater intrusion management in coastal aquifers. *Journal of Hydro-Environment Research*, *29*(November 2019), 45–58. https://doi.org/10.1016/j.jher.2019.11.005

Rojas Gonzalez, S., Jalali, H., & Van Nieuwenhuyse, I. (2020). A multiobjective stochastic simulation optimization algorithm. *European Journal of Operational Research*, *284*(1), 212–226. https://doi.org/10.1016/j.ejor.2019.12.014

Rostami, H., & Khaksar Manshad, A. (2014). Application of evolutionary Gaussian processes regression by particle swarm optimization for prediction of dew point pressure in gas condensate reservoirs. *Neural Computing and Applications*, *24*(3–4), 705–713. https://doi.org/10.1007/s00521-012-1275-9

Roy, D. K., & Datta, B. (2018a). Influence of sea level rise on multiobjective management of saltwater intrusion in coastal aquifers. *Journal of Hydrologic Engineering*, *23*(8), 1–17. https://doi.org/10.1061/(ASCE)HE.1943-5584.0001680

Roy, D. K., & Datta, B. (2018b). Trained meta-models and evolutionary algorithm based multi-objective management of coastal aquifers under parameter uncertainty. *Journal of Hydroinformatics*, *20*(6), 1247–1267. https://doi.org/10.2166/hydro.2018.087

Roy, D. K., & Datta, B. (2019a). Adaptive Management of Coastal Aquifers Using Entropy-Set Pair Analysis–Based Three-Dimensional Sequential Monitoring Network Design. *Journal of Hydrologic Engineering*, *24*(3), 04018072. https://doi.org/10.1061/(asce)he.1943-5584.0001765

Roy, D. K., & Datta, B. (2019b). An Ensemble Meta-Modelling Approach Using the Dempster-Shafer Theory of Evidence for Developing Saltwater Intrusion Management Strategies in Coastal Aquifers. *Water Resources Management*, *33*(2), 775–795. https://doi.org/10.1007/s11269-018-2142-y

Salam, M. A., & Khan, S. A. (2016). Simulation based decision support system for optimization: A case of Thai logistics service provider. *Industrial Management and Data Systems*, *116*(2), 236–254. https://doi.org/10.1108/IMDS-05-2015-0192

Schapire, R. E. (1990). The Strength of Weak Learnability. *Machine Learning*, *5*(2), 197–227. https://doi.org/10.1023/A:1022648800760

Schruben, L. (1986). Simulation optimization using frequenct domain methods. *Winter Simulation Conference*, 366–369. https://doi.org/10.1017/CBO9781107415324.004

Shariatinasab, R., Vahidi, B., Hosseinian, S. H., & Ametani, A. (2008). Optimization of surge arrester's location on EHV and UHV power networks using simulation optimization method. *IEEJ Transactions on Power and Energy*, *128*(12), 1465–1472. https://doi.org/10.1541/ieejpes.128.1465

Sharif, S. A., & Hammad, A. (2019). Developing surrogate ANN for selecting near-optimal building energy renovation methods considering energy consumption, LCC and LCA. *Journal of Building Engineering*, *25*(April), 100790. https://doi.org/10.1016/j.jobe.2019.100790

Shirazi, B., Mahdavi, I., & Mahdavi-Amiri, N. (2011). ICoSim-FMS: An intelligent co-simulator for the adaptive control of complex flexible manufacturing systems. *Simulation Modelling Practice and Theory*, *19*(7), 1668–1688. https://doi.org/10.1016/j.simpat.2011.04.003

Smew, W., Young, P., & Geraghty, J. (2013). Supply chain analysis using simulation, gaussian process modelling and optimisation. *International Journal of Simulation Modelling*, *12*(3), 178–189. https://doi.org/10.2507/IJSIMM12(3)4.239

Song, C., Guan, X., Zhao, Q., & Ho, Y. C. (2005). Machine learning approach for determining feasible plans of a remanufacturing system. *IEEE Transactions on Automation Science and Engineering*, *2*(3), 262–275. https://doi.org/10.1109/TASE.2005.849090

Song, C., Guan, X., Zhao, Q., & Jia, Q. S. (2011). Remanufacturing planning based on constrained ordinal optimization. *Frontiers of Electrical and Electronic Engineering in China*, *6*(3), 443–452. https://doi.org/10.1007/s11460-011-0162-y

Sousa Junior, W. T., Montevechi, J. A. B., Miranda, R. de C., & Campos, A. T. (2019). Discrete simulation-based optimization methods for industrial engineering problems: A systematic literature review. *Computers and Industrial Engineering*, *128*(December 2018), 526–540. https://doi.org/10.1016/j.cie.2018.12.073

Sousa Junior, W. T., Montevechi, J. A. B., Miranda, R. de C., Oliveira, M. L. M. de, & Campos, A. T. (2020). Shop floor simulation optimization using machine learning to improve parallel metaheuristics. *Expert Systems with Applications*. https://doi.org/10.1016/j.eswa.2020.113272

Sousa Junior, W. T., Montevechi, J. A. B., Miranda, R. de C., Rocha, F., & Vilela, F. F. (2019). Economic lot-size using machine learning, parallelism, metaheuristic and simulation. *International Journal of Simulation Modelling*, *18*(2), 205–216. https://doi.org/10.2507/IJSIMM18(2)461

Sreekanth, J., & Datta, B. (2011). Optimal combined operation of production and barrier wells for the control of saltwater intrusion in coastal groundwater well fields. *Desalination and Water Treatment*, *32*(1–3), 72–78. https://doi.org/10.5004/dwt.2011.2680

Steer, K. C. B., Wirth, A., & Halgamuge, S. K. (2012). Decision tree ensembles for online operation of large smart grids. *Energy Conversion and Management*, *59*, 9–18. https://doi.org/10.1016/j.enconman.2012.01.010

Stork, A., Thole, C. A., Klimenko, S., Nikitin, I., Nikitina, L., & Astakhov, Y. (2008). Towards interactive simulation in automotive design. *Visual Computer*, *24*(11), 947–953. https://doi.org/10.1007/s00371-008-0274-4

Storti, B. A., Dorella, J. J., Roman, N. D., Peralta, I., & Albanesi, A. E. (2019). Improving the efficiency of a Savonius wind turbine by designing a set of deflector plates with a metamodel-based optimization approach. *Energy*, *186*, 115814. https://doi.org/10.1016/j.energy.2019.07.144

Tanila, T., Tenhunen, H., & Hirvonen, P. (2020). Value mechanisms in the implementation of intelligent patient flow management system - A multiple case study. *Studies in Health Technology and Informatics*, *270*, 708–712. https://doi.org/10.3233/SHTI200252

Timani, B., & Peralta, R. (2017). Aggregated surrogate simulator for groundwater-surface water management via simulation-optimization modeling: Theory, development and tests. *Environmental Modelling and Software*, *96*, 210–231. https://doi.org/10.1016/j.envsoft.2017.06.014

Vahidi, B., Tavakoli, M. R. B., & Hosseinian, S. H. (2008). Determining arresters best positions in power system for lightning shielding failure protection using simulation optimization approach. *EUROPEAN TRANSACTIONS ON ELECTRICAL POWER*, *20*(3), 255–276. https://doi.org/10.1002/etep

Vapnik, V. N. (1999). *The Nature of Statistical Learning Theory*.

Villarreal-Marroquín, M. G., Mulyana, R., Castro, J. M., & Cabrera-Ríos, M. (2011). Selecting process parameter in injection molding via simulation optimization. *Journal of Polymer Engineering*, *31*(5), 387–395. https://doi.org/10.1515/POLYENG.2011.086

Villarreal-Marroquín, M. G., Svenson, J. D., Sun, F., Santner, T. J., Dean, A., & Castro, J. M. (2013). A comparison of two metamodel-based methodologies for multiple criteria simulation optimization using an injection molding case study. *Journal of Polymer Engineering*, *33*(3), 193–209. https://doi.org/10.1515/polyeng-2013-0022

Vu, K. K., D'Ambrosio, C., Hamadi, Y., & Liberti, L. (2017). Surrogate-based methods for black-box optimization. *International Transactions in Operational Research*, *24*(3), 393–424. https://doi.org/10.1111/itor.12292

Wade, B. M. (2019). Creating surrogate models for an air and missile defense simulation using design of experiments and neural networks. *Journal of Defense Modeling and Simulation*. https://doi.org/10.1177/1548512919877987

Wan, H. P., & Ren, W. X. (2015). Parameter selection in finite-element-model updating by global sensitivity analysis using gaussian process metamodel. *Journal of Structural Engineering*, *141*(6), 1–11. https://doi.org/10.1061/(ASCE)ST.1943-541X.0001108

Wang, H., Chen, L., & Li, E. (2018). Time dependent sheet metal forming optimization by using Gaussian process assisted firefly algorithm. *International Journal of Material Forming*, *11*(2), 279–295. https://doi.org/10.1007/s12289-017-1352-9

Wang, H., Yuan, J., & Ng, S. H. (2020). Gaussian process based optimization algorithms with input uncertainty. *IISE Transactions*, *52*(4), 377–393. https://doi.org/10.1080/24725854.2019.1639859

Wildridge, V., & Bell, L. (2002). How CLIP became ECLIPSE: a mnemonic to assist in searching for health policy/ management information. *Health Information and Libraries Journal*, *19*(1), 113–115. https://doi.org/10.1002/ajpa.21406

Woldemariam, E. T., & Lemu, H. G. (2019). A machine learning based framework for model approximation followed by design optimization for expensive numerical simulation-based optimization problems. *Proceedings of the International Offshore and Polar Engineering Conference*, *1*, 943–950.

Xia, Y., Liu, C., Li, Y. Y., & Liu, N. (2017). A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. *Expert Systems with Applications*, *78*, 225–241. https://doi.org/10.1016/j.eswa.2017.02.017

Xiao, H., Lee, L. H., & Chen, C. H. (2015). Optimal budget allocation rule for simulation optimization using quadratic regression in partitioned domains. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *45*(7), 1047–1062. https://doi.org/10.1109/TSMC.2014.2383997

Xiao, Y., & Watson, M. (2019). Guidance on Conducting a Systematic Literature Review. *Journal of Planning Education and Research*, *39*(1), 93–112. https://doi.org/10.1177/0739456X17723971

Xu, J., Huang, E., Hsieh, L., Lee, L. H., Jia, Q. S., & Chen, C. H. (2016). Simulation optimization in the era of Industrial 4.0 and the Industrial Internet. *Journal of Simulation*, *10*(4), 310–320. https://doi.org/10.1057/s41273-016-0037-6

Xu, L., Zhao, J., Li, C., Li, C., Wang, X., & Xie, Z. (2020). Simulation and prediction of hydrological processes based on firefly algorithm with deep learning and support vector for regression. *International Journal of Parallel, Emergent and Distributed Systems*, *35*(3), 288–296. https://doi.org/10.1080/17445760.2019.1593409

Yadav, B., Ch, S., Mathur, S., & Adamowski, J. (2016). Estimation of in-situ bioremediation system cost using a hybrid Extreme Learning Machine (ELM)-particle swarm optimization approach. *Journal of Hydrology*, *543*, 373–385. https://doi.org/10.1016/j.jhydrol.2016.10.013

Yang, L., & Shami, A. (2020). On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, *415*, 295–316. https://doi.org/10.1016/j.neucom.2020.07.061

Yaohui, L. (2017). A Kriging-based global optimization method using multi-points infill search criterion. *Journal of Algorithms and Computational Technology*, *11*(4), 366–377. https://doi.org/10.1177/1748301817725307

Ye, W., & You, F. (2015). A simulation-based optimization method for solving the integrated supply chain network design and inventory control problem under uncertainty. *Chemical Engineering Transactions*, *45*, 499–504. https://doi.org/10.3303/CET1545084

Yin, H., Fang, H., Wang, Q., & Wen, G. (2016). Design optimization of a MASH TL-3 concrete barrier using RBF-based metamodels and nonlinear finite element simulations. *Engineering Structures*, *114*, 122–134. https://doi.org/10.1016/j.engstruct.2016.02.009

Yousefi, M., & Yousefi, M. (2019). Human resource allocation in an emergency department: A metamodel-based simulation optimization. *Kybernetes*, *49*(3), 779–796. https://doi.org/10.1108/K-12-2018-0675

Yousefi, M., Yousefi, M., Ferreira, R. P. M., Kim, J. H., & Fogliatto, F. S. (2018). Chaotic genetic algorithm and Adaboost ensemble metamodeling approach for optimum resource planning in emergency departments. *Artificial Intelligence in Medicine*, *84*, 23–33. https://doi.org/10.1016/j.artmed.2017.10.002

Zakerifar, M., Biles, W. E., & Evans, G. W. (2011). Kriging metamodeling in multiple-objective simulation optimization. *Simulation*, *87*(10), 843–856. https://doi.org/10.1177/0037549711411964

Zeinali, F., Mahootchi, M., & Sepehri, M. M. (2015). Resource planning in the emergency departments: A simulation-based metamodeling approach. *Simulation Modelling Practice and Theory*, *53*(1), 123–138. https://doi.org/10.1016/j.simpat.2015.02.002

Zhang, C., Osorio, C., & Flötteröd, G. (2017). Efficient calibration techniques for large-scale traffic simulators. *Transportation Research Part B: Methodological*, *97*, 214–239. https://doi.org/10.1016/j.trb.2016.12.005

Zlatinov, M., & Laskowski, G. (2015). Hybrid large-eddy simulation optimization of a fundamental turbine blade turbulated cooling passage. *Journal of Guidance, Control, and Dynamics*, *38*(6), 1292–1297. https://doi.org/10.2514/1.B35713