Proceedings of the 2023 Winter Simulation Conference
C. G. Corlu, S. R. Hunter, H. Lam, B. S. Onggo, J. Shortle, and B. Biller, eds.

ENSEMBLE-BASED INFILL SEARCH SIMULATION OPTIMIZATION FRAMEWORK

José Arnaldo B. Montevechi
João Victor S. Amaral
Rafael C. Miranda
Carlos H. dos Santos

Flávio O. D. Brito
Michael E. F. H. S. Machado

Production Engineering and Management Institute
Federal University of Itajubá
Av. BPS, 1303
Itajubá, MG, 37500-903, BRAZIL

Project and Training Department
FlexSim Brazil
221 Regente Feijó
Campinas, SP, 13013-050, BRAZIL

ABSTRACT

Simulation is widely used in several areas of knowledge, from engineering to biology, including physics and finance. It allows the evaluation of the model's results under different conditions, enabling performance analysis and more assertive decision-making. However, simulation can be computationally intensive, especially when we consider complex models. To deal with this problem, metamodeling has been increasingly used as a simulation optimization technique. In this article, we propose a new adaptive metamodeling method for simulation optimization, which aims to achieve better results using fewer experiments. This method combines machine learning and metaheuristic techniques, allowing the identification of the most important regions of the search space, which can be explored more efficiently to obtain optimal solutions. The results achieved in a manufacturing problem show that the proposed method presents a significant improvement in the achieved objective function value, in comparison with the conventional benchmark method, without compromising the simulation execution time.

1 INTRODUCTION

The Discrete event simulation (DES) is a technique widely used in several areas of knowledge, from manufacturing to services, including logistics, hospitals, retail, military, among others. DES aims to computationally replicate real systems, in operation or designed, allowing the responses of this system to be evaluated under different conditions, enabling performance analysis and more assertive decision-making (Santos et al. 2021).

DES can assist in various decision-making processes, such as examining bottlenecks on a production line, determining the appropriate number of machines on a workstation, staffing a medical facility, testing different layout configurations, and scheduling assignments in a workshop, among other applications. These purposes usually involve several decision variables and an objective function (FO) that must be achieved, resulting in numerous scenarios to be evaluated. In this case, Fu (2002) emphasizes the importance of using simulation optimization techniques (OvS) to find the scenario that optimizes the problem. However, running the simulation can be computationally intensive, especially when using complex models, which might make the OvS process impractical due to the time required. To deal with this problem, many authors suggest elevating the problem to another level of abstraction, the so-called metamodeling.

Metamodeling aims to identify and estimate the relationship between inputs and outputs of the simulation model, forming a simplified model (metamodel) that is used to evaluate possible solutions in the optimization process. According to Parnianifard et al. (2019), replacing the simulation model with the metamodel allows reducing the total time invested in optimization. In this way, several works proposed the creation of more accurate and efficient metamodeling methods, capable of outlining good results with few

979-8-3503-6966-3/23/$31.00 ©2023 IEEE

simulation executions. Some examples include the works of (Moghaddam and Mahlooji 2017; Pang et al. 2022; Parnianifard et al. 2018).

Amaral et al. (2022) point out that metamodeling approaches can be classified into fixed experimental design (or one-shot), and incremental strategy (or infill-search). In the fixed design, a single experimental base with size *w* is selected at the beginning of the project and the metamodel is trained using only this set. For the incremental strategy, an initial experimental base is selected, the metamodel is trained and optimized and, based on the acquisition criterion, new points are added to the training base. Then, the metamodel is retrained with the new experimental base. This cycle continues until the stopping criterion is reached. The incremental strategy is the focus of this work.

Several studies can be found in the literature involving this approach to simulation optimization, where the use of metamodels based on Kriging and Polynomial Regression is very common (Bharaj et al. 2015; Ng and Yin 2012; Wang and Ierapetritou 2017; Yaohui 2017). Thus, this study aims to propose a new metamodeling method, combining Design of Experiments, Ensemble models (Bagging), Gradient-Boosted Trees (GBT), hyper-parameter optimization and Genetic Algorithm (GA).

Furthermore, this article compares the proposed method with Efficient Global Optimization (EGO), an incremental strategy method based on Kriging that is widely used in the literature (Gu 2021; He et al. 2021; Raponi et al. 2021). In this case, we aim to validate the results found by the proposed algorithm. Moreover, we adopted a real case study, a simulation model that acts as a digital twin for weekly resource allocation in a small/medium-sized factory in the textile sector.

The rest of this article is divided into 4 more Sections. Section 2 presents the theoretical framework on which this work is based. Section 3 presents the proposed method, while Section 4 discusses its application in a case study. Section 5 concludes the work and proposes directions for future research.

2 BACKGROUND

Miranda et al. (2017) describe OvS as the optimization of an objective function (OF) subject to constraints, which is evaluated through a stochastic simulation. OvS refers to the process of identifying the best input values for the variables of a simulated system, evaluating the solutions through a loop between the optimization algorithm and the simulation model. The OvS study field has evolved significantly in the last decades, with the development of several algorithms, software, and applications. However, computational time is still a challenge for OvS.

According to Amaran et al. (2016), the general formulation for an OvS problem consists of finding the minimum value of the objective function $\mathbb{E}_{\omega}[f(x,y,\omega)]$, subject to the constraints $\mathbb{E}_{\omega}[g(x,y,\omega)] \leq 0$ and $h(x,y) \leq 0$, where $x_l \leq x \leq x_u$, $x \in \mathbb{R}^n, y \in \mathbb{D}^m$. The function f is evaluated through simulation with continuous x or discrete y inputs, subject to a vector of random numbers ω. The constraints are defined by the values of the g function that are evaluated at each simulation. The problem may contain other constraints (represented by h) that do not involve random variables, as well as constraints linked to decision variables.

However, Oliveira et al. (2017) emphasize that OvS problems, considering complex systems with very large solution space, the computational time required for the optimization algorithm to converge to a good result might be long. To overcome this challenge, Barton (2009) mentions that researchers have developed specialized methods for OvS, including ranking and selection, heuristics and metaheuristics, random search and metamodeling.

According to De La Fuente and Smith (2017), the complexity of the studied system directly influences the time required to perform the optimization. To obtain good results in a reasonable time, the authors suggest the use of metamodeling, an approach that consists of developing a representative model of the simulation model. The metamodel can capture the relationship between the decision variables and the simulation outputs, providing an approximation of the objective function in a much shorter time than the simulation. Sousa Junior et al. (2019) also highlight the effectiveness of the metamodel in reducing the optimization runtime.

According to Amaral et al. (2022a), frameworks for metamodeling can be divided into two categories: the fixed experimental design strategy (one-shot) and the incremental design strategy (infill-search). The fixed sampling strategy consists of taking a single set of samples of size w at the beginning of the project, in which the metamodel is trained exclusively with these samples. On the other hand, the incremental strategy starts with an initial sample of data of size ξ, in which the metamodel is trained and optimized to determine the accuracy and direction of the region in which the optimum lies. After that, the database is incremented with a new set of samples of size δ_i in each iteration of the algorithm. The metamodel is retrained with a base size $\xi_{i+1} = \xi_i + \delta_i$, where i is the number of iterations of the algorithm.

In many works, the algorithm responsible for recursive and adaptive learning is known as EGO. In this algorithm, Kriging is used as a metamodel and the choice of points to be added in each iteration is usually determined by the acquisition function called Expected Improvement (EI), as established in Equation (1).

$$E[I(\boldsymbol{x})] = \left(f_{min} - \hat{f}(\boldsymbol{x})\right)\Phi\left(\frac{f_{min} - \hat{f}(\boldsymbol{x})}{\sigma}\right) + \sigma\phi\left(\frac{f_{min} - \hat{f}(\boldsymbol{x})}{\sigma}\right) \tag{1}$$

Where Φ is the normal cumulative distribution function and ϕ is the normal probability density function. f_{min} is the smallest value observed in the training base and $\hat{f}(\boldsymbol{x})$ is the prediction of the metamodel (Kriging) for the point $\boldsymbol{x}$ and σ is the standard deviation of the prediction. If we evaluate the derivatives of the function EI with respect to $\hat{f}(\boldsymbol{x})$ and σ, we will notice that the value of EI is greater with lower $\hat{f}(\boldsymbol{x})$ and higher s. Therefore, when maximizing the EI function, it will tend to find places with the best trade-off between local search (smaller $\hat{f}(\boldsymbol{x})$) and global search (larger s).

With the development of computing resources, several algorithms were used to improve the execution speed, accuracy and reliability of metamodeling. In their systematic literature review, Amaral et al. (2022a) concluded that the main metamodels used in OvS are Kriging, Polynomial Regression, Neural Networks, GBT, Random Forest, Radial Basis Function, and Support Vector Machine. Additionally, the works proposed by Amaral et al. (2022b) and Amaral et al. (2022c) compared several metamodels in the one-shot strategy in manufacturing problems, concluding that the GBT algorithm obtained the best performance. Based on these results, and others highlighted in the literature (Ganjisaffar et al. 2011; Louk and Tama 2023), the method proposed in this work will use the Bagging algorithm with GBT as base-learner as metamodel.

According to Friedman (2002), GBT is a machine learning (ML) algorithm widely used in several applications, including regression and classification problems. GBT is a form of boosting algorithm that builds a set of decision trees iteratively, where each new tree is built to correct the errors of the previous ones. The algorithm starts by building a single decision tree and, at each iteration, a new decision tree is added to the set. The new tree is built by fitting the negative gradient of the current model's loss function, which updates the model's predictions. The final model is obtained by combining the predictions of all decision trees in the set (Zhang and Haghani 2015).

GBT has several advantages over other ML algorithms. It can handle numerical and categorical features and it is robust to outliers and missing data. Additionally, GBT can capture complex nonlinear relationships between features and the target variable, making it suitable for high-dimensional data. GBT has been successfully applied in various fields such as finance, healthcare, and marketing to solve a wide range of problems including fraud detection, disease diagnosis, and customer churn prediction (Praveena and Jaiganesh 2017; Xia et al. 2017; Zhang and Haghani 2015). However, GBT is computationally intensive and requires careful tuning of hyper-parameters to achieve optimal performance (Amaral et al. 2022b).

The Bagging algorithm (Bootstrap Aggregating) is an Ensemble technique that involves the combination of multiple ML models, called base-learners, trained on randomly sampled data sets according to the bootstrapping technique (Breiman 1996). Let $\boldsymbol{D}$ be the original dataset with n data points and let B be the number of models we want to train. Let $\boldsymbol{D_i}$ be the *ith* subset of $\boldsymbol{D}$, with m data points randomly sampled with replacement from $\boldsymbol{D}$. For each b = 1, 2, ..., B, B_b model is trained with subset $\boldsymbol{D_i}$ using some supervised learning algorithm, which in our case is GBT (Hastie et al. 2008).

The Bagging model error is measured by the out-of-sample R^2 (OOB), which is calculated using the training samples that were not selected for a specific model during the bagging process. Thus, for each OOB sample, the bagging algorithm uses all models that were not sampled during training to make a prediction. This results in a set of predictions for each OOB sample on which R^2 is calculated. More details on OOB can be found in (Ramosaj and Pauly 2019; Schonlau and Zou 2020).

According to Hastie et al. (2008), after training, the Bagging model predictions $\hat{f}(\boldsymbol{x})$ are defined by aggregating the predictions of each base-learner, defined by $\hat{f}^{*b}(\boldsymbol{x})$, generating a final prediction which is more stable and generally more accurate than any individual model. The final prediction and standard deviation of the prediction of the Bagging model are defined by Equations (2) and (3).

$$\hat{f}(\boldsymbol{x}) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(\boldsymbol{x}) \tag{2}$$

$$\sigma = \frac{1}{B}\sum_{b=1}^{B} \left(\hat{f}(\boldsymbol{x}) - \hat{f}^{*b}(\boldsymbol{x})\right)^2 \tag{3}$$

According to Louk and Tama (2023), when GBT is used as a base-learner for Bagging, it can bring several benefits. First, Bagging can help reduce the overall variance of the model, improving the accuracy of predictions. Furthermore, the sequential nature of GBT means that each subsequent model can focus on correcting the previous model's errors, which can lead to reduced forecast bias. Finally, Bagging can help improve model robustness by reducing the impact of outliers or imbalanced data on the original dataset. Therefore, the association of these two techniques allows the reduction of the variance and bias of the model (Ganjisaffar et al. 2011).

3 MATERIALS AND METHOD

Figure 1 presents the method proposed in this work. This method is based on five main techniques: Latin Hypercube Design (LHD), Hyper-parameters Optimization, GBT, Bagging, and Genetic Algorithm for optimization of the acquisition function (Balanced Expected Improvement). The implementation of this method was carried out in the Python language (version 3.0).

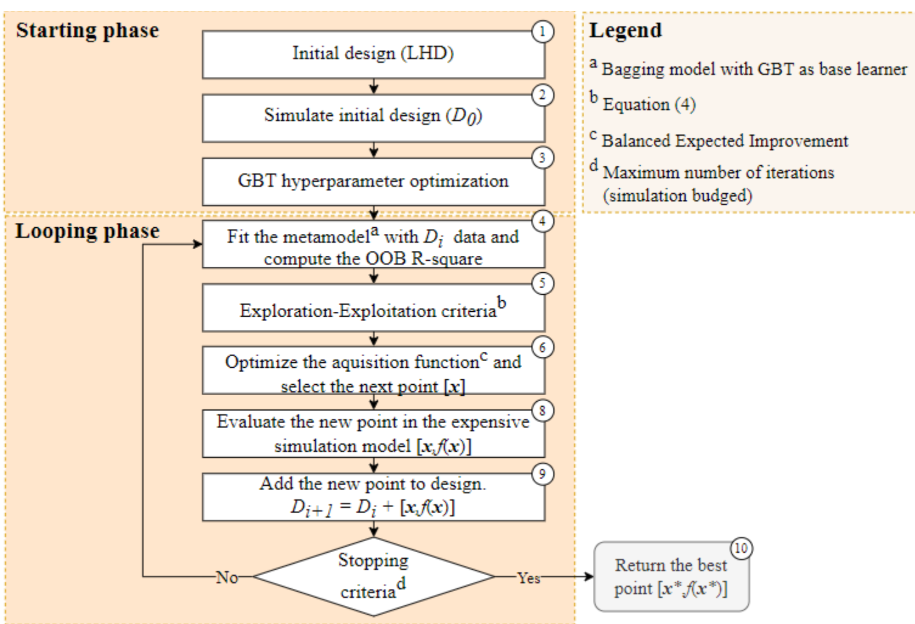

Figure 1: Proposed optimization method.

This method includes two macro phases, the initialization phase and the recursive phase. Step 1 of the initialization phase comprises the generation of the initial array ($\boldsymbol{D_0} = [\boldsymbol{X}, f(\boldsymbol{X})]$) for training the metamodel. For this, the initial arrangement will be outlined by the Design of Experiments technique known as Latin Hypercube Design (LHD). The LHD is a very popular stratified sampling method in metamodeling studies. Given n samples and s variables, the LHD divides the region of each variable into n disjoint intervals with equal probability, and then the sample space is selected by constructing the n x s matrix, where the columns are randomly selected from the permutation of [1, ..., n] and each line corresponds to a cell of a hyper-rectangle. After building the matrix, a point from each of the cells is sampled (Luo et al. 2019). The suggested sample size for the initial LHD array is $2x$ to $10x$, where x is the problem dimension (Bharaj et al. 2015; Wang and Ierapetritou 2017). In Step 2, the experimental arrangement $\boldsymbol{X}$ is evaluated in the simulation model, generating the response $f(\boldsymbol{X})$ for the points, forming the initial training base of the metamodel.

Step 3 optimizes the GBT hyper-parameters through the Genetic Algorithm (GA). GA is an optimization technique based on the theory of natural selection that has been successfully used in hyperparameter optimization. In GA, the search space is composed of candidate solutions, represented by chromosomes composed of genes that encode the possible solutions of the problem (Li et al. 2010). The purpose of GA is to combine the genes to generate new chromosomes with better FO values. For hyper-parameter optimization and GBT training, GA uses the k-fold cross-validation technique to calculate the error associated with each combination of parameters of the ML algorithm. The k-fold cross-validation divides the training data into k equal and random parts, using k – 1 parts for training and calculating the error on the rest of the data. Each set of parameters has its error calculated k times, and in each interaction a different part k_i' is selected for testing, with i = {1, ..., k}. Finally, the error is calculated on the average of the k parts (Bergmeir and Benítez 2012).

In Step 4, the recursive phase of the method begins and comprises the training of the metamodel. In this step, the Bagging algorithm with optimized GBT (base-learner) is recursively trained with the training base $\boldsymbol{D_i}$ of size ξ, in which a point is added to each iteration i of the algorithm, $\xi_{i+1} = \xi_i + 1$. In this step, the R^2 (OOB) of the metamodel is also calculated, which will be used in the next step. Step 5 is related to the Exploration-Exploitation criteria draw. This criterion is a fundamental trade-off that arises in decision-making considering uncertain or unknown situations. It refers to the choice between exploring new options or taking advantage of known options. Exploration involves searching for new options, collecting information and experimenting with different alternatives, that is, regions where the metamodel prediction error is high. This approach is useful when the environment is unpredictable and there is a need to learn about new opportunities or find optimal solutions. On the other hand, Exploitaition involves maximizing current knowledge by choosing the best option in known regions of the solution space, i.e., best value of $\hat{f}(\boldsymbol{x})$ (Ajdari and Mahlooji 2014).

In this step, the objective is to define the value of the binary variable λ, which assumes the value 0 for Exploration and 1 for Exploitation. Considering ρ a uniformly distributed random number, such that $\rho \sim$ Uniform(a,b), where a and b are the lower and upper bounds for the draw, respectively. In this work, the value of a was defined as the minimum value between 0.85 and R^2, and b as 1, representing the minimum desirable R^2 to start an Exploitation search and the maximum value to allow Exploration, respectively. The value of λ is selected based on the drawn value for ρ and the R^2 calculated in the previous step. Therefore, a higher R^2 value means a greater probability of Exploitation (λ=1), according to Equation (4).

$$\lambda = \begin{cases} 1, & if\ \rho < R^2 \\ 0, & Otherwise \end{cases} \tag{4}$$

Step 6 comprises the optimization of the acquisition function, whose objective is to select the best point to be simulated and added to the training base in the next iteration of the algorithm. The acquisition function proposed in this work is a variation of the EI, called Balanced Expected Improvement (BEI), defined by Equation (5).

$$BEI(\boldsymbol{x}) = (1-\lambda)\sigma\phi\left(\frac{f_{min} - \hat{f}(\boldsymbol{x})}{\sigma}\right) + \lambda\hat{f}(\boldsymbol{x}) \tag{5}$$

Where ϕ represents the probability density function, f_{min} is the smallest value observed in the training base, $\hat{f}(\boldsymbol{x})$ v and σ are the forecast and standard deviation of the metamodel forecast for the point $\boldsymbol{x}$, defined by Equations (2) and (3), respectively. The first term of Equation (5) describes the Exploration component, and the second term represents the Exploitation component. Note that the value of λ found in the previous step is the criterion that defines whether the acquisition function will weigh Exploration or Exploitation. The goal is to find the point $\boldsymbol{x}$ that maximizes the value of BEI, and to achieve this goal, GA is used as a search engine. As GA parameters, a population size of 10 individuals was used, the tournament as a selection criterion, the type of mutation as uniform, the mutation probability of 0.1, the number of generations fixed at 100 and maximum number of generations without improvement equal to 10.

In Step 7, the best solution found by the GA is evaluated in the simulation model, returning the value of $f(\boldsymbol{x})$ for this point. After this step, the new point is added to the metamodel training base, $\boldsymbol{D_{i+1}} = \boldsymbol{D_i} + [\boldsymbol{x}, f(\boldsymbol{x})]$, and steps 4 – 8 are repeated until the stopping criterion is reached. In this article, the stopping criterion was defined as the maximum number of iterations. When it is reached, the algorithm is terminated and the solution $\boldsymbol{x}^*$ with the best value of $f(\boldsymbol{x}^*)$ is returned as the solution to the problem.

4 RESULTS AND DISCUSSION

4.1 Case Study

To explore the effectiveness of the proposed method, this section presents its application in a real resource allocation problem in the textile industry. The case represents a small/medium-sized factory of a fast-fashion manufacturing sector, whose main characteristic is the rapid change in demand and product mix, short cycle time, high volatility, low predictability and high level of impulsive buying and price competition. To meet these sector requirements, the key business factors comprise a flexible production system and decision-making based on effective and fast-responsive tools. The company works with weekly demand forecasts, to which on Friday the manager receives next week's demand and must decide on the allocation of productive resources (machines) and staff to meet such demand and boost the profits of the factory. We focused on the main production line of the factory. The manager might lease equipment and employees from other production lines of the company; however, it may result in additional operational costs. More details about this case can be found in Santos et al. (2021). The DES model, presented in Figure 2, was developed and validated in the FlexSim® simulation software (version 22.2.3).

Figure 2: DES model.

The purpose of the model is to provide support for this decision-making. Through OvS, it seeks to find the allocation of resources that optimizes the profit of the factory. The decision variables, $\boldsymbol{x} = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7\ x_8]^{\mathrm{T}}$, are the size of the team dedicated to the group of tasks type 1 and type 2 (x_1 and x_2), and the amount of equipment of different types allocated to the process (x_3 to x_8). Given the lower bounds, $\boldsymbol{L} = [2\ 2\ 1\ 2\ 2\ 2\ 2\ 2]^{\mathrm{T}}$, and upper bounds, $\boldsymbol{U} = [10\ 10\ 3\ 10\ 10\ 10\ 10\ 10]^{\mathrm{T}}$, of the decision variables subject to the weekly unit cost $\boldsymbol{C} = [800\ 800\ 500\ 500\ 500\ 300\ 350\ 300]^{\mathrm{T}}$. The profit function is given by Equation (6).

$$L(\boldsymbol{x}) = \sum_{j=1}^{J} \psi_j \upsilon_j - \sum_{i=1}^{8} x_i C_i \tag{6}$$

Where $L(\boldsymbol{x})$ represents the profit, ψ_j is the weekly production achieved by the model, and υ_i is the contribution margin for product j. The objective function comprises the maximization of the expectation for the profit function, $\max f(\boldsymbol{x}) = \mathbb{E}_{\omega}[L(\boldsymbol{x}, \omega)]$, which can be obtained by averaging $L(\boldsymbol{x})$ of the simulation model replications, subject to distinct ω random seeds. For this model, 15 replications were adopted.

To evaluate its effectiveness, the proposed method was applied to optimize this OvS problem, given a limited simulation budget. In addition, the EGO method, implemented by the python package Surrogate Modeling Toolbox (Bouhlel et al. 2019) was chosen as a benchmark, given its proven effectiveness in several studies and since it is one of the most used in studies involving metamodeling (Amaral et al. 2022a). For comparison purposes, in both algorithms, the stopping criterion was defined as the maximum number of simulations, with 50% destined to generate the initial training base.

For this evaluation, two tests were performed, the first with a total of 80 experiments (80 simulations with 15 replications each one) and the second with 160 experiments. Furthermore, given the stochastic character of these methods, the optimization process was repeated 10 times for each method, in order to evaluate their responses and the achieved robustness. The following section presents the results of the algorithms in each budget.

4.2 Test 1 Results

This section presents the optimization results of the problem presented in Section 4.1 considering a simulation budget of 80 experiments. Figure 3 (a) presents the mean and error for the convergence graph obtained with the EGO method. At the end of the 80 experiments, the EGO obtained an average result of \$47,827.07 and standard deviation of \$936.42, reaching results that varied between \$46,900.4 and \$49,512.0 of profit in the worst and best repetition of the algorithm, respectively.

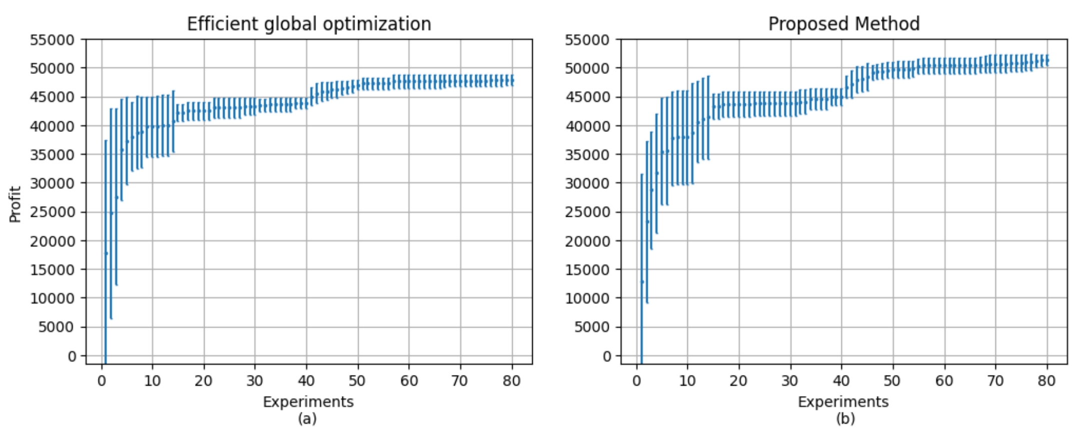

Figure 3: Convergence graph for EGO (a) and the proposed method (b) with 80 experiments.

Figure 3 (b) presents the results of the 10 replicates of the proposed method considering the budget of 80 experiments. On average, the proposed method achieved a profit of $51,322.95 and a standard deviation of $974.89, with results varying between $49,332.71 and $52,516.80, as shown in Figure 4 (a). Comparing the results of the methods using the non-parametric Mann-Whitney statistical test (Fay and Proschan 2010), it appears that the proposed method presents significantly better results than the EGO method (p-value < 0.001), with a difference of $3,627.20 in the median and $3,495.88 on average. Regarding the time required to perform the optimization, there was no significant difference between both methods, with the EGO requiring an average of 1,554 seconds and the proposed method requiring 1,535 seconds, as shown in Figure 4 (b).

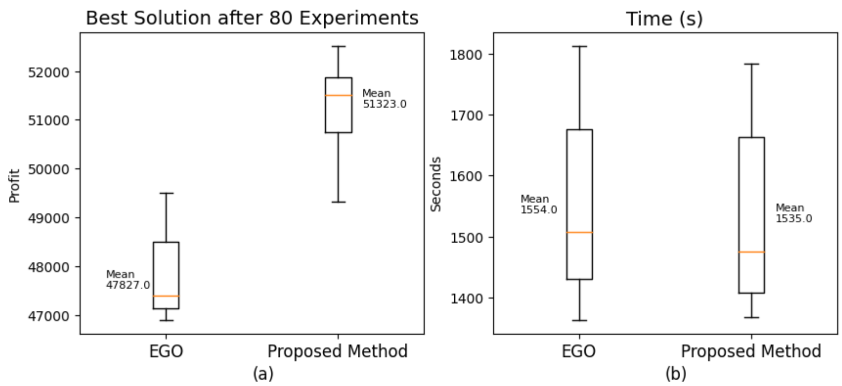

Figure 4: Boxplot for profit value achieved after 80 experiments (a) and time spent on optimization (b).

4.3 Test 2 Results

This section discusses the results obtained with the optimization by the EGO and the proposed method considering a simulation budget of 160 experiments (with 15 replicates each one). Figure 5 shows the convergence graph for both methods.

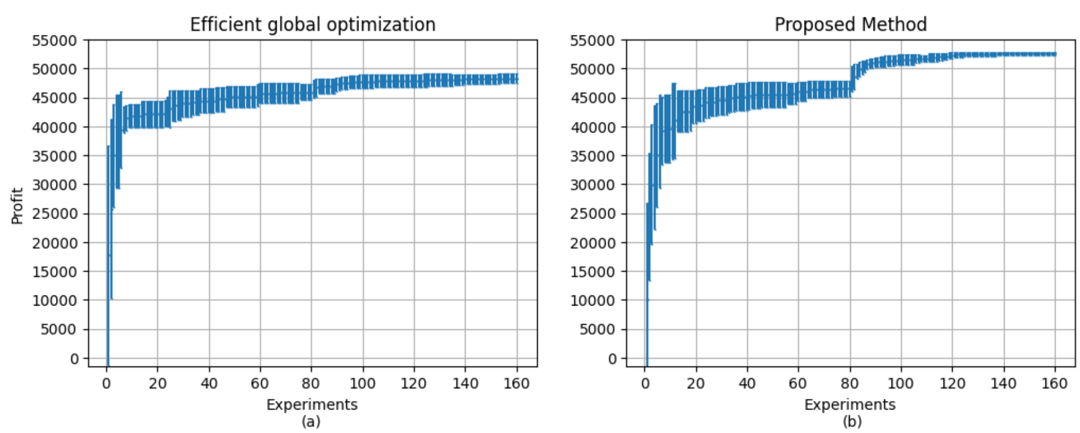

Figure 5: Convergence graph for EGO (a) and the proposed method (b) with 160 experiments.

Considering the new experiment budget as a stopping criterion, the EGO method reached, on average, a profit of $48,262.04, varying between $47,148.80 and $49,017.60 (Figure 6 (a)), demanding an average time of 3,351 seconds (Figure 6 (b)). On the other hand, the proposed method obtained an average profit of $52,558.13, varying between $51,955.20 and $52,750.40, demanding an average time of 3,374 seconds. We noted that the proposed method reached a result significantly superior to the EGO, confirmed by the Mann-Whitney test (p-value < 0.001), with a difference of $3,903.64 for the median and $4,296.09 for the mean. Furthermore, an expressive reduction in the variability of the results is highlighted, demonstrating the robustness of the method.

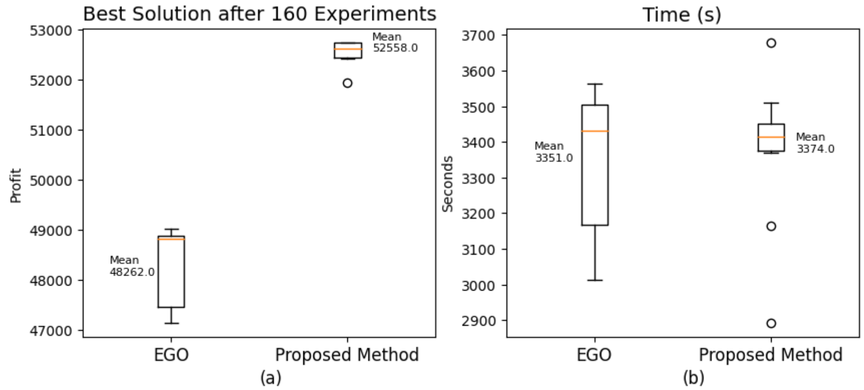

Figure 6: Boxplot for profit value achieved after 160 experiments (a) and time spent on optimization (b).

5 CONCLUSION

Discrete event simulation aims to computationally replicate real systems and evaluate performance under different conditions. However, running the simulation might be computationally intensive, especially when using complex models, making the optimization process impractical due to the time required. In this sense, metamodeling emerges as a viable solution to reduce the total time involved in the optimization.

This work proposes an optimization method based on metamodeling that combines Design of Experiments, Bagging model, GBT, hyper-parameter optimization and Genetic Algorithm. The proposed method is formulated in ten macro steps, as detailed in Section 3, and aims to train the metamodel based on an incremental sampling strategy, where, at each iteration, a new point is selected to be simulated and incorporated into the metamodel training database. This new point is selected based on the optimization (via GA) of the BEI acquisition function, which expresses the trade-off between Exploration and Exploitaition based on the predictions of the previously trained metamodel.

To exemplify the applicability of the proposed method, this article presents an OvS case in the fast-fashion sector. The case was modeled via DES and aims at the optimal allocation of productive resources, in order to meet the weekly demand predicted. The problem aims to optimize the factory's profit and presents eight discrete decision variables, comprising a search space with 14,348,907 possible solutions. To evaluate its performance, the proposed method was used to optimize this problem considering two simulation budgets, 80 experiments and 160 experiments. The results demonstrated the effectiveness of the proposed method in comparison with another metamodeling method commonly used in the literature, the EGO, which was defined as the benchmark of this work. Both methods were submitted to 10 replicates, in order to analyze their results and their robustness against randomness.

For the first test, with a budget of 80 experiments, the proposed method reached an average profit of 51,322.95, while the EGO reached an average profit of 47,827.07. In the second test, with 160 experiments, the proposed method reached an average profit of 52,558.13 compared to 48,262.04 achieved by the EGO. Considering the non-parametric test for Mann-Whitney medians, in both tests the proposed method was superior to the benchmark (p-value < 0.001), demanding the same computational time.

The proposed method demonstrated consistent results for the presented case, which represents a typical resource allocation problem in DES, which exhibits peculiar characteristics, such as high stochasticity, process interdependence, nonlinearity, and possible discontinuity of the objective function. These features require the use of metamodels with high learning capacity, such as those used in this work. It is noteworthy that EGO is an optimization method based on metamodeling that is widely used in the literature and with excellent results in a wide variety of problems. Therefore, it is recommended that future research evaluate and compare the proposed method in new DES problems and/or other optimization problems.

This article proposes the use of an adaptive acquisition function, called BEI. However, future works can explore other acquisition functions within the proposed method, as well as EI, Probability of Improvement, Surrogate-based, Lower-bound Confidence, among others. Another extension would be the evaluation of other algorithms to act as base-learners for Bagging, e.g., Support Vector Machine, Neural Networks and Extreme Gradient Boosting. Considering that the initial arrangement has a strong influence on the optimization results, the study of better techniques for its creation and/or the ideal size of the initial training base may be an important contribution to the research area.

REFERENCES

Ajdari, A., and H. Mahlooji. 2014. "An Adaptive Exploration-Exploitation Algorithm for Constructing Metamodels in Random Simulation Using a Novel Sequential Experimental Design". *Communications in Statistics: Simulation and Computation* 43(5):947–68. doi: https://doi.org/10.1080/03610918.2012.720743.

Amaral, J. V. S., J. A. B. Montevechi, R. C. Miranda, and W. T. S. Junior. 2022a. "Metamodel-Based Simulation Optimization: A Systematic Literature Review". *Simulation Modelling Practice and Theory* 114:102403. doi: https://doi.org/10.1016/j.simpat.2021.102403.

Amaral, J. V. S., R. C. Miranda, J. A. B. Montevechi, C. H. Santos, and G. T. Gabriel. 2022b. "Metamodeling-Based Simulation Optimization in Manufacturing Problems: A Comparative Study". *International Journal of Advanced Manufacturing Technology* 120(7–8):5205–24. doi: https://doi.org/10.1007/s00170-022-09072-9.

Amaral, J. V. S., R. C. Miranda, J. A. B. Montevechi, C. H. Santos, and A. F. Silva. 2022c. "Data Envelopment Analysis for Algorithm Efficiency Assessment in Metamodel-Based Simulation Optimization". *International Journal of Advanced Manufacturing Technology* 121(11–12):7493–7507. doi: https://doi.org/10.1007/s00170-022-09864-z.

Amaran, S., N. V. Sahinidis, B. Sharda, and S. J. Bury. 2016. "Simulation Optimization: A Review of Algorithms and Applications". *Annals of Operations Research* 240(1):351–80. doi: https://doi.org/10.1007/s10479-015-2019-x.

Barton, R. R. 2009. "Simulation Optimization Using Metamodels". In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, 230-238. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Bharaj, G., D. I. W. Levin, J. Tompkin, Y. Fei, H. Pfister, W. Matusik, and C. Zheng. 2015. "Computational Design of Metallophone Contact Sounds". *ACM Transactions on Graphics* 34(6):1–13. doi: https://doi.org/10.1145/2816795.2818108.

Bouhlel, M. A., J. T. Hwang, N. Bartoli, R. Lafage, J. Morlier, and J. R. R. A. Martins. 2019. "A Python Surrogate Modeling Framework with Derivatives". *Advances in Engineering Software* 135. doi: https://doi.org/10.1016/j.advengsoft.2019.03.005.

Fay, M. P., and M. A. Proschan. 2010. "Wilcoxon-Mann-Whitney or T-Test? On Assumptions for Hypothesis Tests and Multiple Interpretations of Decision Rules". *Statistics Surveys* 4:1–39. doi: https://doi.org/10.1214/09-SS051.

Friedman, J. H. 2002. "Stochastic Gradient Boosting". *Computational Statistics and Data Analysis* 38(4):367–78. doi: 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. doi: https://doi.org/10.1287/ijoc.14.3.192.113.

Ganjisaffar, Y., R. Caruana, and C. V. Lopes. 2011. "Bagging Gradient-Boosted Trees for High Precision, Low Variance Ranking Models". In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, July 25-29, 2011, Beijin, China, 85–94.

Gu, J. 2021. "An Efficient Multiple Meta-Model-Based Global Optimization Method for Computationally Intensive Problems". *Advances in Engineering Software* 152:102958. doi: https://doi.org/10.1016/j.advengsoft.2020.102958.

Hastie, T., R. Tibshirani, and J. Friedman. 2008. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. New York: Springer.

He, Y., J. Sun, P. Song, and X. Wang. 2021. "Variable-Fidelity Expected Improvement Based Efficient Global Optimization of Expensive Problems in Presence of Simulation Failures and Its Parallelization". *Aerospace Science and Technology* 111:106572. doi: https://doi.org/10.1016/j.ast.2021.106572.

De La Fuente, R., and R. Smith. 2017. "Metamodeling a System Dynamics Model: A Contemporary Comparison of Methods". In *Proceedings of the 2017 Winter Simulation Conference*, edited by W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, 1926-1937. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Louk, M. H. L., and B. A. Tama. 2023. "Dual-IDS: A Bagging-Based Gradient Boosting Decision Tree Model for Network Anomaly Intrusion Detection System". *Expert Systems with Applications* 213. doi: https://doi.org/10.1016/j.eswa.2022.119030.

Miranda, R. C., J. A. B. Montevechi, A. F. Silva, and F. A. S. Marins. 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):673–81. doi: https://doi.org/10.1016/j.ejor.2017.04.016.

Moghaddam, S., and H. Mahlooji. 2017. "A New Metamodel-Based Method for Solving Semi-Expensive Simulation Optimization Problems". *Communications in Statistics: Simulation and Computation* 46(6):4795–4811. doi: https://doi.org/10.1080/03610918.2015.1134567.

Ng, S. H., and J. Yin. 2012. "Bayesian Kriging Analysis and Design for Stochastic Simulations". *ACM Transactions on Modeling and Computer Simulation* 22(3). doi: https://doi.org/10.1145/2331140.2331145.

Oliveira, M. L. M., J. A. B. Montevechi, A. F. Pinho, and R. C. Miranda. 2017. "Using Hybrid Simulation to Represent the Human Factor in Production Systems". *International Journal of Simulation Modelling* 16(2):263–74. doi: https://doi.org/10.2507/IJSIMM16(2)7.378.

Pang, M., E. Du, C. A. Shoemaker, and C. Zheng. 2022. "Efficient, Parallelized Global Optimization of Groundwater Pumping in a Regional Aquifer with Land Subsidence Constraints". *Journal of Environmental Management* 310:114753. doi: https://doi.org/10.1016/j.jenvman.2022.114753.

Parnianifard, A., A. S. Azfanizam, M. K. A. Ariffin, M. I. S. Ismail, and N. A. Ebrahim. 2019. "Recent Developments in Metamodel Based Robust Black-Box Simulation Optimization: An Overview". *Decision Science Letters* 8(1):17–44. doi: https://doi.org/10.5267/j.dsl.2018.5.004.

Parnianifard, A., A. S. Azfanizam, M. K. A. Ariffin, M. I. S. Ismail, M. R. Maghami, and C. Gomes. 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):1471–79. doi: https://doi.org/10.1166/jctn.2018.7379.

Praveena, M., and V. Jaiganesh. 2017. "A Literature Review on Supervised Machine Learning Algorithms and Boosting Process". *International Journal of Computer Applications* 169(8):32–35. doi: https://doi.org/10.5120/ijca2017914816.

Ramosaj, B., and M. Pauly. 2019. "Consistent Estimation of Residual Variance with Random Forest Out-Of-Bag Errors". *Statistics and Probability Letters* 151:49–57. doi: https://doi.org/10.1016/j.spl.2019.03.017.

Raponi, E., D. Fiumarella, S. Boria, A. Scattina, and G. Belingardi. 2021. "Methodology for Parameter Identification on a Thermoplastic Composite Crash Absorber by the Sequential Response Surface Method and Efficient Global Optimization". *Composite Structures* 278:114646. doi: https://doi.org/10.1016/j.compstruct.2021.114646.

dos Santos, C. H., G. T. Gabriel, J. V. S. Amaral, J. A. B. Montevechi, and J. A. Queiroz. 2021. "Decision-Making in a Fast Fashion Company in the Industry 4.0 Era: A Digital Twin Proposal to Support Operational Planning". *International Journal of Advanced Manufacturing Technology* 116(5–6):1653–66. doi: https://doi.org/10.1007/s00170-021-07543-z.

Schonlau, M., and R. Y. Zou. 2020. "The Random Forest Algorithm for Statistical Learning". *Stata Journal* 20(1):3–29. doi: https://doi.org/10.1177/1536867X20909688.

Sousa Junior, W. T., J. A. B. Montevechi, R. C. Miranda, F. Rocha, and F. F. Vilela. 2019. "Economic Lot-Size Using Machine Learning, Parallelism, Metaheuristic and Simulation". *International Journal of Simulation Modelling* 18(2):205–16. doi: https://doi.org/10.2507/IJSIMM18(2)461.

Wang, Z., and M. Ierapetritou. 2017. "A Novel Surrogate-Based Optimization Method for Black-Box Simulation with Heteroscedastic Noise". *Industrial and Engineering Chemistry Research* 56(38):10720–32. doi: https://doi.org/10.1021/acs.iecr.7b00867.

Xia, Y., C. Liu, Y.Y. Li, and N. Liu. 2017. "A Boosted Decision Tree Approach Using Bayesian Hyper-Parameter Optimization for Credit Scoring". *Expert Systems with Applications* 78:225–41. doi: https://doi.org/10.1016/j.eswa.2017.02.017.

Yaohui, Li. 2017. "A Kriging-Based Global Optimization Method Using Multi-Points Infill Search Criterion". *Journal of Algorithms and Computational Technology* 11(4):366–77. doi: https://doi.org/10.1177/1748301817725307.

Zhang, Y., and A. Haghani. 2015. "A Gradient Boosting Method to Improve Travel Time Prediction". *Transportation Research Part C: Emerging Technologies* 58:308–24. doi: https://doi.org/10.1016/j.trc.2015.02.019.

AUTHOR BIOGRAPHIES

JOSE ARNALDO BARRA MONTEVECHI is a Titular Professor of the Production Engineering and Management Institute at the Federal University of Itajubá, in Brazil. He holds the degrees of Mechanical Engineer from the Federal University of Itajubá, M.Sc. in Mechanical Engineer from the Federal University of Santa Catarina, and Doctorate of Engineering from Polytechnic School of the University of São Paulo. His research interest includes Operational Research, Simulation, and Economic Engineering. His email address is montevechi@unifei.edu.br.

JOÃO VICTOR SOARES DO AMARAL is a Ph.D. Student in Industrial Engineering at the Federal University of Itajubá, in Brazil. He holds the degrees of Industrial Engineer from the Federal University of São João del-Rei and M.Sc. in Industrial Engineer from the Federal University of Itajubá. His research interest includes Simulation, Machine Learning, Digital Twins, and Simulation-based optimization. His email address is joao.amaral@unifei.edu.br.

RAFAEL DE CARVALHO MIRANDA is a Professor of the Production Engineering and Management Institute at the Federal University of Itajubá, inBrazil. He holds the degrees of Production Engineer, M.Sc., and Doctorate in Production Engineering from the Federal University of Itajubá. His research interests include Operational Research, Simulation, Simulation-based Optimization, and Metamodeling. His email address is rafael.miranda@unifei.edu.br.

CARLOS HENRIQUE DOS SANTOS is a Ph.D. Student in Industrial Engineering at the Federal University of Itajubá, in Brazil. His bachelor's and master's degrees in Industrial Engineering from the Federal University of Itajubá. His research interest includes Simulation, Industry 4.0, Digital Twins, and Simulation-based optimization. His email address is chenrique.santos@unifei.edu.br.

FLÁVIO OLIVEIRA DE BRITO is the Technical Manager and Partner at FlexSim Brazil. He has been working with Discrete Event Simulation since 2010, acting in many projects at several areas. He holds a degree in Industrial Engineer and Marster's in Production and Manufacturing Engeneering. His research interests include Project Planning and Simulation overall, focusing in Hybrid Simulation and modeling real systems. His email address is flavio.obrito@gmail.com.br.

MICHAEL E. F. H. S. MACHADO is the Business Manager at Flexsim Software Corporate Brazil. He has been working with Discrete Event Simulation since 2007, leading Brazilian Flexsim Software operations and business. He holds a Master and Certificater in Business Adminstration at Insper and IBMEC. His research interests include leadership and simulation, focusing in optimization. His email address is michael.machado@flexsimbrasil.com.br.