JCEC
THE JOURNAL OF ENGINEERING AND EXACT SCIENCES

OPEN ACCESS

The Journal of Engineering and Exact Sciences – jCEC, Vol. 09 N. 10 (2023)
journal homepage: https://periodicos.ufv.br/jcec
eISSN: 2527-1075
ISSN: 2446-9416

Using Gaussian Processes for Metamodeling in Robust Optimization Problems

Uso de Processos Gaussianos para a Metamodelagem em Problemas de Otimização Robusta

Article Info:
Article history: Received 2023-09-03 / Accepted 2023-11-20 / Available online 2023-12-29
doi: 10.18540/jcecvl9iss10pp17809

Claudemir Mota da Cruz
ORCID: https://orcid.org/0000-0001-7264-6220
State University of Santa Cruz - DCEX, Ilhéus, BA, Brazil
Polytechnic Institute, Rio de Janeiro State University - Nova Friburgo, RJ, Brazil
E-mail: cmcruz@uesc.br
Fran Sérgio Lobato
ORCID: https://orcid.org/0000-0002-7401-4718
Federal University of Uberlândia, Faculty of Chemical Engineering - Uberlândia, MG, Brazil
E-mail :fslobato@ufu.br
Gustavo Barbosa Libotte
ORCID: https://orcid.org/0000-0002-4583-6026
Polytechnic Institute, Rio de Janeiro State University - Nova Friburgo, RJ, Brazil
E-mail: gustavolibotte@iprj.uerj.br

Resumo
Este artigo propõe uma abordagem baseada em Processos Gaussianos para a construção de metamodelos para problemas de otimização robusta que busca diminuir o esforço computacional requerido para quantificar incertezas. A abordagem é aplicada em dois casos: um problema de *benchmark* de baixa dimensão e um de projeto estrutural, de alta dimensão, que consiste em minimizar a massa de uma estrutura formada por barras de diferentes materiais e diâmetros, submetida a cargas pontuais em diferentes locais. Os casos são modelados como problemas de otimização robusta, onde a função objetivo é estimada por um Processo Gaussiano e o procedimento de otimização é conduzido empregando-se uma meta-heurística populacional. Os resultados indicam que a abordagem proposta é eficaz na redução do número de avaliações de função objetivo necessárias para a obtenção de uma solução robusta, não havendo diferenças estatísticas significativas na qualidade das soluções alcançadas
Palavras-chave: Processos. Gaussianos. Metamodelagem. Otimização. Robusta.

Abstract
This article proposes an approach based on Gaussian Processes for building metamodels for robust optimization problems that seek to reduce the computational effort required to quantify uncertainties. The approach is applied to two cases: a low-dimensional benchmark problem and a high-dimensional structural design, which consists of minimizing the mass of a structure formed by bars of different materials and diameters, subjected to point loads in different locations. The cases are modeled as robust optimization problems, where the objective function is estimated by a Gaussian Process and the optimization procedure uses a population meta-heuristic. The results indicate that the proposed approach is effective in reducing the number of objective function evaluations required to obtain a robust solution, with no significant statistical differences in the quality of solutions achieved.
Keywords: Gaussian. Process. Metamodels. Optimization. Robust

1. Introduction

Ideal solutions to optimization problems are often not viable in practice, because the results found in nominal problems can be in highly nonlinear regions. The problem parameters can vary due to uncertainties arising from various sources, such as properties of materials and variations in environmental conditions, among others. Thus, the system may no longer operate satisfactorily, according to Tsutsui, *et al.*, 1996. In these cases, the solution must be robust, that is, it must be able to maintain system performance, even when subject to small external disturbances.

The concept of robustness is closely linked to the name of Genichi Taguchi, who states that robustness is the state where the performance of the technology, product, or process is minimally sensitive to factors that cause variability (whether in manufacturing or in the user s environment) and aging at the lowest unit cost of manufacturing" (Park , *et al.*, 2006, citing Taguchi, 1987).

However, the computational cost of evaluating robustness can become prohibitive when the problem requires significant computational effort, as is the case with finite element analysis problems (Yang *et al*, 2022). This difficulty can be circumvented using metamodels, which approximate the objective function or the constraints of the problem at an acceptable computational cost to achieve optimal results like the original problem (Jiang *et al*, 2020).

In this work, a methodology based on Gaussian Processes is proposed for metamodeling during the robustness evaluation process. This is done to reduce the number of objective function evaluations and thus decrease the computational cost of the procedure. At the same time, the results achieved using metamodels are analyzed to assess their deviation from the solutions achieved with the original models. Therefore, the joint objective of the proposed methodology is to use metamodeling to reduce computational cost and obtain solutions equivalent to those obtained with the original model in the context of robust optimization.

The rest of this article is organized as follows: In Section 2, a brief review on genetic algorithms, Gaussian processes, and robust optimization is presented. In Section 3, the proposed methodology is described. In Section 4, the results obtained with the application of the proposed methodology in a low-dimensional benchmark problem and a high-dimensional structural engineering problem are presented. Finally, in Section 5, the conclusions and perspectives for future work are presented.

2. Regression with Gaussian Processes, Genetic Algorithms and Robustness

This section briefly describes Gaussian process regression, the use of genetic algorithms in the optimization process, and the concept of robustness.

2.1 Regression with Gaussian Processes

A Gaussian Process is a probabilistic model that describes the dependence between random variables of a collection, so that any finite subset of them follows a joint Gaussian distribution (Rasmussen and Williams, 2006). A Gaussian process can be used to perform regression, which consists of estimating a function from a set of input and output data.

Consider a function $f: \mathcal{X} \rightarrow \mathcal{Y}$ that maps an input $\mathcal{X}$ space to an output space $\mathcal{Y}$. Suppose this function is unknown (of the black box type: only some input and output pairs are known, but not the rule that relates them), or it is very costly to evaluate at all points of the input space, so only a set of data is available $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ are vectors. The objective is to estimate f in a set of points, $\mathcal{X}^* = \{x_1^*, x_2^*, \dots, x_m^*\}$, using regression with Gaussian Processes (Rasmussen and Williams, 2006).

A Gaussian Process, which we denote as $f(x) \sim \mathrm{GP}(m(x), k(x, x'))$, is completely determined by its mean and covariance. The mean and covariance are defined respectively as $m(x) = \mathbb{E}[f(x)]$ and $k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$ where $\mathbb{E}$ is the expectation operator and k is called the kernel function or covariance function.

In the regression process, it is generally assumed that the output of a function is given by $y = f(x) + \epsilon$, where ϵ is a Gaussian noise with mean zero and variance σ^2, that is, $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$, (Rasmussen e Williams, 2006).

A kernel function is a crucial component of the Gaussian Process, as it determines how the points in the input set relate to each other. One of the most common choices is the Radial Basis Function (RBF), also known as the Gaussian function, which is expressed as follows:

$$k(x, x') = \sigma_f^2 \exp\left(-\frac{1}{2l^2}||x - x'||^2\right) \tag{1}$$

Here, σ_f^2 is the variance of the function and l is the length scale. These values are known as hyperparameters and are adjusted during the metamodeling process, in order to maximize the likelihood of the model.
The joint distribution between the training points (observed points) and the unobserved points is given by:

$$\begin{bmatrix} y \\ f^* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K + \sigma_n^2 I & K_* \\ K_*^T & K_{**} \end{bmatrix}\right), \tag{2}$$

where K is the covariance matrix between the training points, K_* is the covariance matrix between the training points and the unobserved points, K_{**} is the covariance matrix between the unobserved points, and I is the identity matrix. With this, the predicted value for the unobserved points is given by:

$$f^* \sim \mathcal{N}(K_*^T(K + \sigma_n^2 I)^{-1} y, K_{**} - K_*^T(K + \sigma_n^2 I)^{-1} K_*) \tag{3}$$

In other words, the prediction at a point x_* and its covariance can be given respectively by:

$$m(x_*) = K_*^T(K + \sigma_n^2 I)^{-1} y \tag{4}$$

and

$$k(x_*) = K_{**} - K_*^T(K + \sigma_n^2 I)^{-1} K_* \tag{5}$$

The detailed development can be consulted in Rasmussen and Williams (2006), and Wackernagel (2003).

2.2 Genetic Algorithms

Genetic Algorithms are indeed useful in the optimization of complex problems, because they allow for the search for solutions in large search spaces, simulating the process of natural evolution. They combine the idea of survival of the fittest individual with the exchange of information between individuals (Goldberg, 1989). Here we present a brief description of how genetic algorithms operate.

At first, a population of individuals is generated randomly and assessed using the objective function. Selection for reproduction is based on the objective function evaluation. The reproductive process employs genetic operators, including crossover and mutation. Crossover merges two chosen individuals to produce a new one, while mutation randomly modifies a selected individual. The individuals generated by both operators are once again evaluated by the objective function, and the fittest are chosen for the next generation. This defines the selection process by favoring the fittest individuals. The process repeats until a stopping criterion is met. Additional details can be found in Goldberg (1989).

2.3 Robustness

In the context of optimization, robustness is associated with solutions that are minimally sensitive to external disturbances, in the context of a problem subject to uncertainties. These uncertainties can be caused by several factors, such as measurement errors, geometric inaccuracies, modeling errors, roundoff errors, among others. Therefore, the optimal solution of a problem that does not consider robustness (the so-called nominal problems) can be unfeasible or even useless when applied in the practice of a project. In some cases, it may not be convenient to follow the optimal solution if it is a very acute point, that is, when small disturbances at the optimal point cause large variations in the value of the objective function. This applies especially to areas where it is important to have a certain safety against variations in environmental conditions, such as the adjustment of the parameters of aerospace or nuclear energy control systems, according to Tsutsui, *et al*. (1996).

Consider the effective mean approach in calculating robust solutions. For a problem of minimizing a function f, a solution x_* is said to be robust if it minimizes the effective mean function, defined in relation to a δ-neighborhood of x_*, where δ is a robustness parameter (Deb and Gupta (2006)). The effective mean function is defined as:

$$f^{\text{eff}}(x) = \frac{1}{|\mathcal{B}_\delta(x)|} \int_{\xi \in \mathcal{B}_\delta(x)} f(\xi) d\xi. \tag{6}$$

However, often the calculation of the equation is unfeasible, and therefore, an approximation is sought for the effective mean function, given by:

$$f^{\text{eff}}(x) \approx \frac{1}{N} \sum_{i=1}^{N} f(\xi_i), \tag{7}$$

where $\xi_i \in B_\delta(x)$ and N is the number of samples in the δ-neighborhood of x .

3. Metamodeling of Robust Optimization Problems

One of the main challenges in analyzing robustness through the effective mean approach is the need to evaluate the objective function for each of the samples in the neighborhood of a candidate solution. This can be computationally infeasible, depending on the complexity of the function involved. Therefore, it is crucial to find efficient methods to approximate the effective mean function, especially for complex optimization problems.

To reduce the computational cost of robustness analysis, it is proposed to use metamodels for the calculation of the values of the effective mean function. This approach can significantly reduce the computational cost in terms of the number of evaluations of the objective function during the optimization process, while achieving a solution as close as possible to that of the corresponding problem in which metamodels are not used.

3.1 Hybrid Genetic Algorithm with Gaussian Processes

The proposed methodology consists of the following stages:

- **Initial Metamodel Training:** The algorithm initializes the candidate solution population using the Latin Hypercube strategy (HCL) and evaluates each member of the population according to the objective function. With the obtained input and output points, Gaussian Processes-based regression model is constructed (trained).
- **Initialization:** In the second stage of the algorithm, the population from the previous stage is used as the initial population. The regression model, trained in the previous stage, serves as the objective function used in the evaluation of the effective mean function. Here, the effective mean function is the function to be optimized.
- **Exploration:** In the third stage, a test set is derived from the new individuals created in the mutation and crossover stage. If the error between the metamodel and the original function in these individuals is arbitrarily greater than 20%, the objective function is evaluated in all

individuals in the current population, and the metamodel is updated. Otherwise, the individuals in the test set are added to the training set and the metamodel is updated. Subsequently, the updated regression model is used as the objective function to evaluate the robustness of the solution candidates through the effective mean approach. This process is repeated for each generation until a stopping criterion related to the model is met.

- **Refinement:** Once such a stopping criterion is met, the fourth stage begins. In this stage, the algorithm relinquishes the use of the surrogate model and begins employing the original objective function to evaluate the individuals of the population in the calculation of the effective mean function, thus initiating the refinement phase. The refinement phase is arbitrarily executed for 10% of the number of generations used by the metamodel. The flowchart of the hybrid genetic algorithm operation is shown in Figure 1.

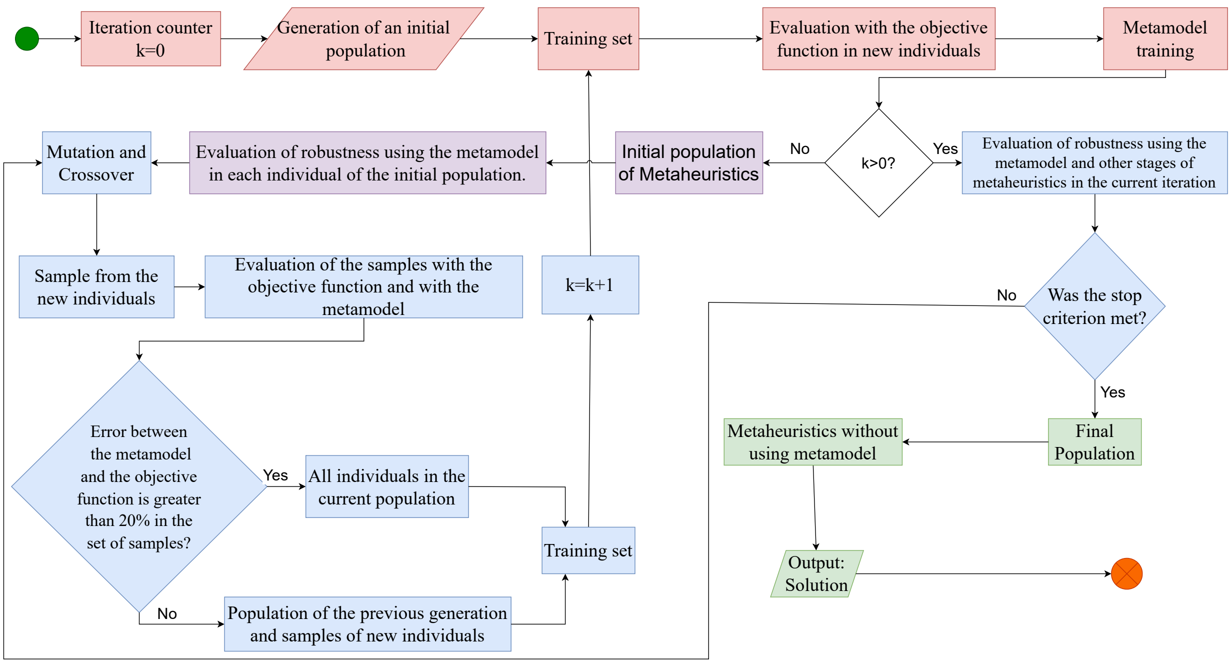

Figure 1: The flowchart of the hybrid genetic algorithm.

The refinement criterion is used to ensure that the metamodel is not used to determine the final robust solution. This is because the metamodel approximates the objective function and can produce solutions that are less accurate than the original model. Therefore, the metamodel is used in the exploratory phase of the algorithm, and the original objective function is used in the refinement phase (Baquela and Oliveira, 2019).

4 Results and Discussion

In this section, the proposed approach for finding robust solutions to optimization problems is evaluated. Initially, a two-dimensional problem is solved to illustrate the behavior of the obtained results. Additionally, an engineering problem is analyzed, aiming to verify the efficiency of the proposed approach in a more challenging problem.

For the execution of the algorithm, the following parameters were used: number of samples used to calculate robustness $N = 50$, maximum number of generations $G_{max} = 11{,}000$, probabilities of mutation and crossover $P_m = P_c = 0.1$, and percentage in relation to the total number of generations during which the genetic algorithm is executed using the metamodel for calculating robustness $G_p = 0.9$. As a stop criterion for the use of the metamodel for the calculation of robustness, a tolerance of 10^{-6} for the difference between the best individual in two subsequent generations was used. Complementarily, a tolerance of 10^{-8} was used for the difference between the best objective function value of two successive iterations.

4.1 Simple Benchmark Problem

Initially, let $f: R^2 \rightarrow R$ be a function given by:

$$f(\boldsymbol{x}) = -\exp\left(-\frac{(x_1+2,5)^2+(x_2+2,5)^2}{0,2^2}\right) \quad -0.8\exp\left(-\frac{(x_1+2)^2+(x_2+1)^2}{0,4^2}\right) + \exp\left(\frac{(x_1+x_2)^2}{100}\right) \tag{8}$$

The constrained optimization problem is given by:

$$\underset{\boldsymbol{x}}{\text{minimize}}\, f(\boldsymbol{x}) \tag{9}$$

Subject to $gi(x) \leq 0$, for $i = 1,2,3$, where

$$g_1(\boldsymbol{x}) = \left(x_1^2 + x_2^2 + 2rx_1\right)^2 - 4r^2\left(x_1^2 + x_2^2\right) + 2 \tag{10}$$

$$g_2(\boldsymbol{x}) = \left(x_1^2 + x_2^2\right)^3 - 4ax_1^2x_2^2 \tag{11}$$

$$g_3(\boldsymbol{x}) = -\left[(x_1 + 1)^2 + x_2^2 - 2r(x_1 + 1)\right]^2 + 4r^2\left[(x_1 + 1)^2 + x_2^2\right] - 2 \tag{12}$$

Additionally, the search space is bounded by

$$-5 \leq x_1, x_2 \leq 0 \tag{13}$$

This problem is multimodal, and the global minimum is located at $x^g = (-2.5, -2.5)$, with $f(x^g) \approx 0.284025$. In addition, there is a local minimum at $x^l = (-2, -1)$, with $f(x^l) \approx 0.294174$. If we consider a robustness level of 10%, evaluating the effective mean function

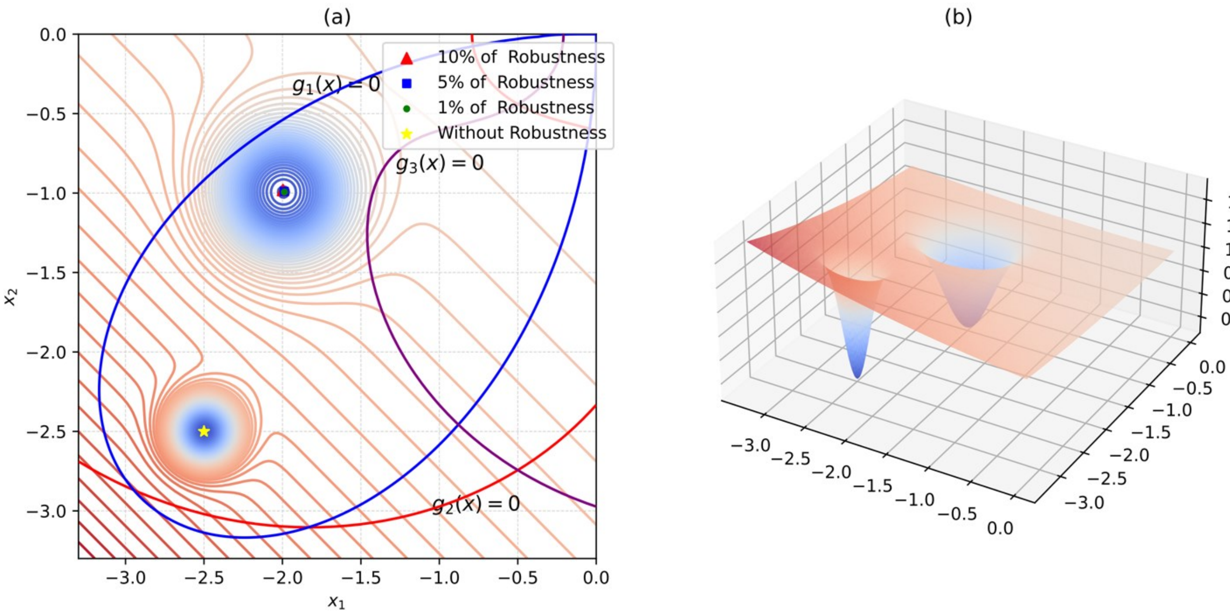

Figure 2 - (a) The feasible region of the problem, along with the level curves of the objective function. (b) A graph of the objective function.

directly, without using the proposed methodology, we obtain $f^{\text{eff}}(x^g) \approx 0{,}871970$ and $f^{\text{eff}}(x^l) \approx 0{,}371026$. This shows that the point x^l is more robust than the point x^g, thus being a more appropriate solution considering uncertainties.

Following the methodology proposed in the previous section, and assuming a robustness level of 10%, the robust point obtained is $\boldsymbol{x} = (-1.996647, -0.980849)$, with $f^{\text{eff}}(\boldsymbol{x}) \approx 0.373839$, which demonstrates that the optimal robust occurs at $\boldsymbol{x^l} = (-2; -1)$. For this level of robustness, 39,620 evaluations of the objective function were necessary without the use of metamodeling, whereas 33,436 evaluations of the objective function with the use of metamodeling, which represents a reduction of 15.6% in the number of evaluations of the objective function, demonstrating the effectiveness of the proposed methodology. Figures 2(a) and 2(b) illustrate the feasible region of the problem, the contour plots of the objective function, and the robust and non-robust optimal points for the different levels of robustness achieved using the proposed methodology.

4.2 - Problem of optimizing a truss bridge

This problem involves finding the optimum geometry of a truss bridge structure, considering uncertainties in the model parameters, to minimize the mass of the structure, maximize the safety factor, and limit the maximum deflection. It was modified from a problem presented by Chen (2022). Figure 3 shows the geometry of the truss bridge, comprising 21 rods and 12 nodes. The design variables are the material of the rods (aluminum, titanium, or steel), the cross sectional area of the rods, and the position of the free nodes in the structure. Free nodes are those that can have their cartesian coordinates (x, z) changed, while fixed nodes have their positions predefined. In this problem, the free nodes are n_2, n_5, n_7, n_9 and n_{11} , and the fixed nodes are $n_1, n_3, n_4, n_6, n_8, n_{10}$ and n_{12} . Therefore, the problem has 52 design variables, 21 relating to the material of the rods, 21 relating to the cross sectional area of the rods, and ten relating to the position of the free nodes.

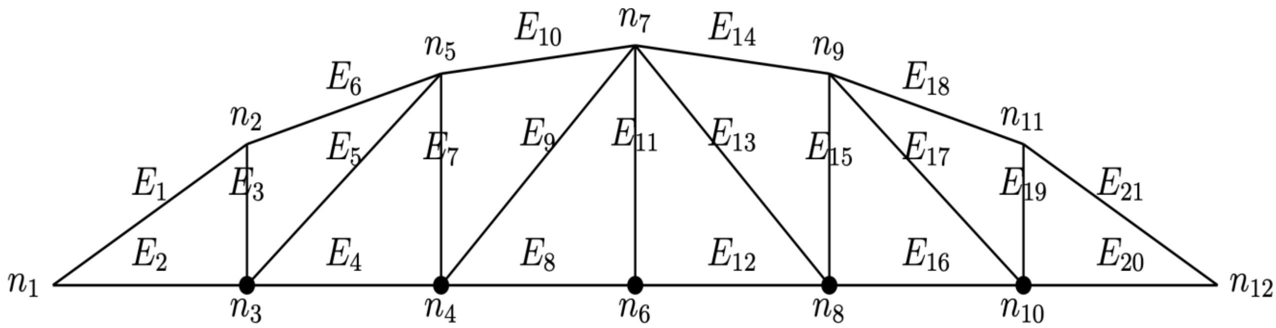

Figure 3 - A truss bridge with weights at points $n_3, n_4, n_6, n_8,$and n_{10}**.**
Modified from Chen (2022).

The objective function to be maximized is given by:

$$fitness(D, SF_{min}, M) \quad = dfit(D) + sfit(SF_{min}) + 3.5 \times mfit(M) \tag{14}$$

With

$$dfit(D) = 1 - \frac{D}{D_{ref}} \tag{15}$$

$$sfit(SF_{min}) = \begin{cases} \frac{SF_{min}}{SF_{target}} - 0.5, if\ SF_{min} < SF_{target} \\ 1 - \frac{SF_{min-}SF_{target}}{SF_{target}}, if\ SF_{min} > SF_{target} \end{cases} \tag{16}$$

$$mfit(M) = 1 - \frac{M}{M_{ref}} \tag{17}$$

where D represents the maximum displacement of the structure and, D_{ref} is a constant that denotes the reference displacement used to calculate the fitness. On the other hand, $SF_{\min}$ corresponds to the minimum safety factor of the structure, which is a critical indicator of system safety. SF_{target} is another essential constant, representing the minimum desired safety factor for the structure. As for the mass of the structure, it is denoted by M, being the total sum of the component masses. M_{ref} is a constant that refers to the reference mass, used as a basis for evaluating fitness. The value of $fitness$ is calculated as the total fitness factor, incorporating the considerations of displacement, safety factor, and mass to evaluate the overall effectiveness of the structure in relation to the established criteria.

It is important to emphasize that D, SF_{min} , and M depend on the design variables, which are the coordinates of the free nodes, the cross sectional area of the rods, and the materials of the rods. Obtaining D, SF_{min} , and M is done through structural analysis, and their calculations are beyond the scope of this work. For more details about structural analysis, please refer to Soriano (2005) and Chen (2022).

The algorithm was executed 100 times, taking 11,000 as the maximum number of generations in each execution, and the results were compared to those obtained using only the genetic algorithm without metamodeling, employing the same parameters.

In Figures 4, 5, and 6, the distribution of the number of evaluations of the objective function performed in each case are presented, considering all executions performed for the different levels of robustness. It is very important to emphasize that these results include the number of evaluations required for the calculation of robustness, the construction of the metamodel, and the refinement at each generation, that is, all the procedures that can impact the computational cost of the proposed methodology.

In Figure 4, the number of evaluations of the objective function to obtain the solution of the truss problem is presented, compared to the approach in which metamodeling is adopted, considering robustness level equal to 10%. It is possible to observe in Figure 4(a) that the use of the metamodel significantly reduced the number of evaluations of the objective function by about ten

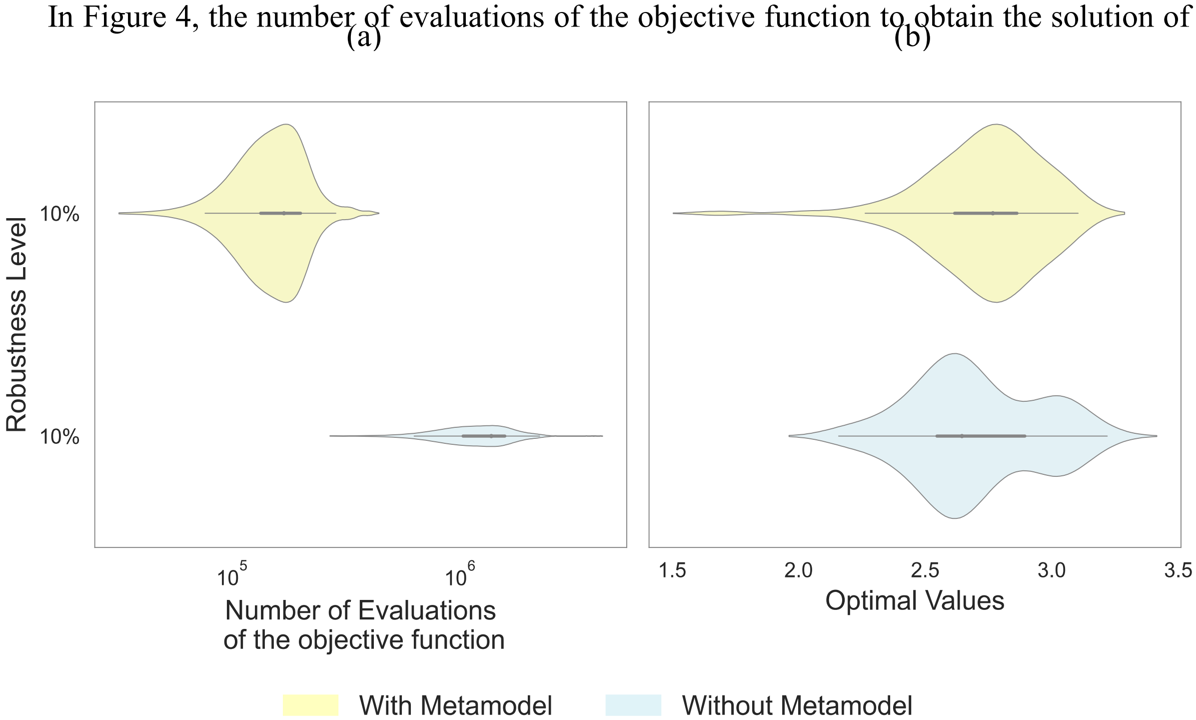

Figure 4 - Comparison between the numbers of evaluations of the objective function for the truss problem using the metamodel and without the metamodel for the calculation of the level of robustness at 10%.

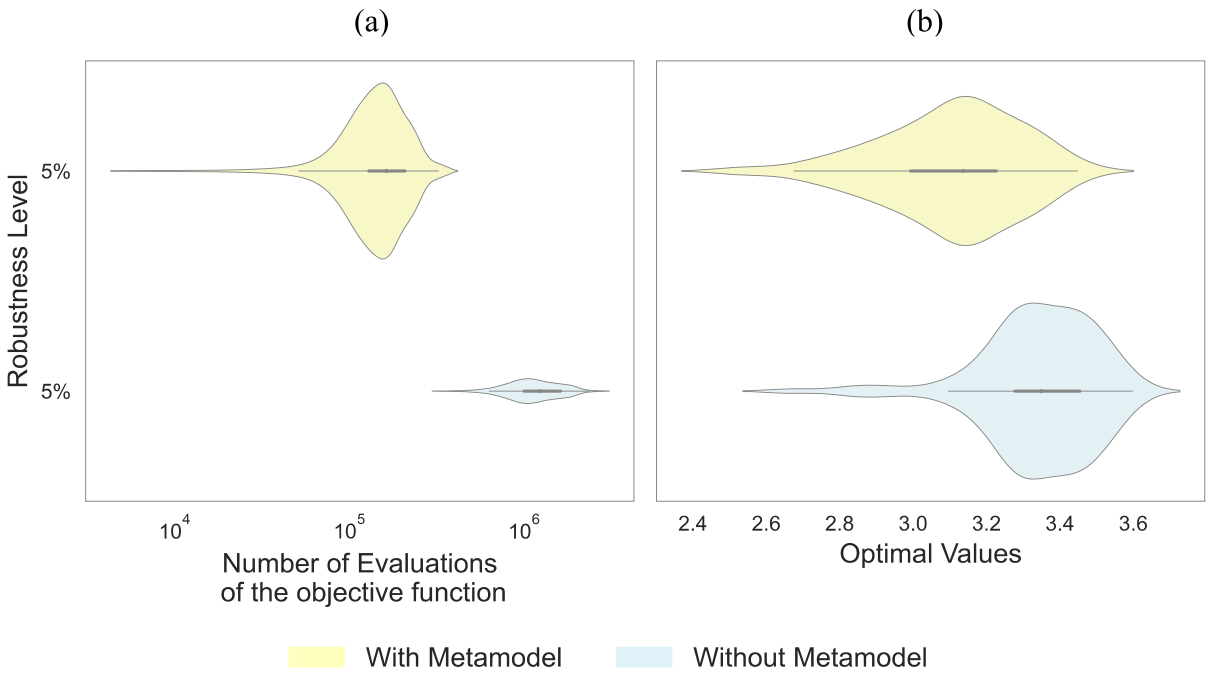

Figure 5 - Comparison between the numbers of evaluations of the objective function for the truss problem using the metamodel and without the metamodel for the calculation of the level of robustness at 5%.

times in relation to the typical strategy (without metamodel). In addition, in Figure 4(b), the optimal values obtained in both cases represent are in quite good agreement.

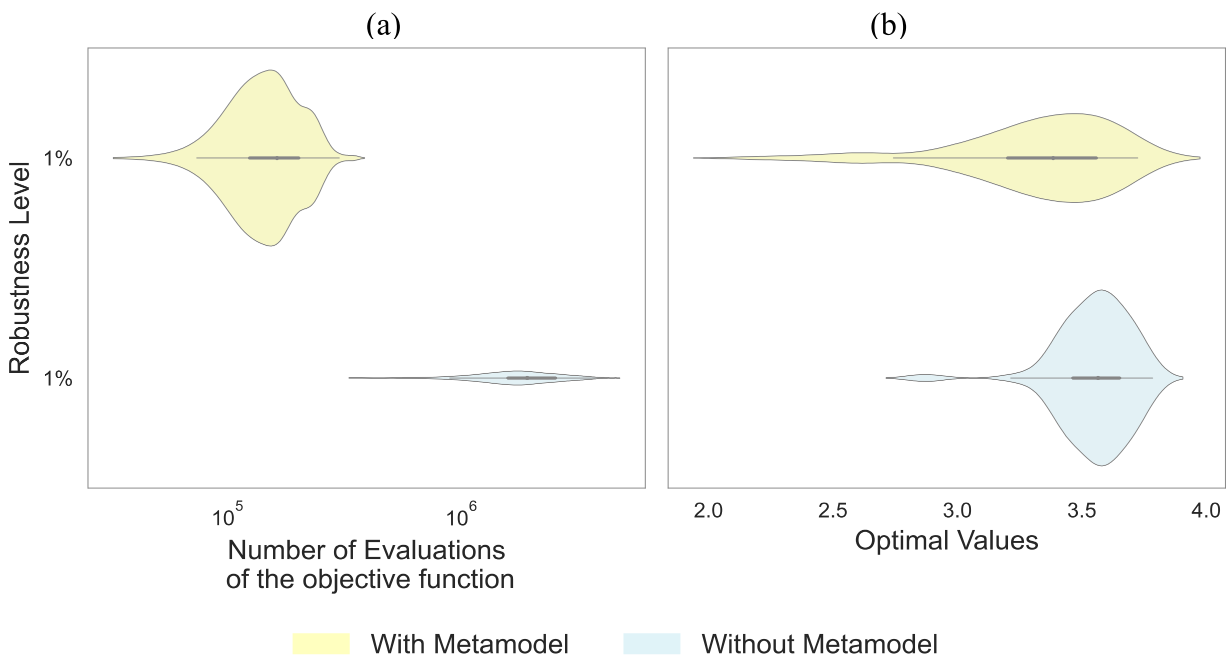

Figure 6 - Comparison between the numbers of evaluations of the objective function for the truss problem using the metamodel and without the metamodel for the calculation of the level of robustness at 1%.

Figures 5 and 6 show that, despite the significant reduction in the number of objective function evaluations when using metamodeling, no significant statistical differences are observed in the results achieved when the original model is replaced by the metamodel. This allows for more generations in the refinement phase as needed, which corroborates the advantages of using metamodeling.

5 Conclusion

In this article, we propose a Gaussian process-based metamodeling approach to estimate the objective function for calculating robustness at different levels. We apply our methodology to two problems, dimensionally distinct, to demonstrate its computational efficiency in terms of reducing the number of objective function evaluations. Our approach proved to be adequate, decreasing the number of objective function evaluations by more than tenfold in the high-dimensional problem without compromising the quality of the robust solution.

One possible improvement to our methodology is to change the stop criterion for the refinement phase to enable greater proximity between the solutions with and without metamodeling. This could further improve the quality of the robust solution.

Another promising direction for future research is to explore the use of metamodels to estimate robustness in multi-objective optimization problems. This could be done by integrating reliability techniques in the search for robust solutions. We could also explore the use of other metamodels, such as neural networks, to estimate robustness in optimization problems.

Acknowledgements

The authors gratefully acknowledge the financial support provided by State University of Santa Cruz and Rio de Janeiro State University. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

References

Baquela, E. G., Olivera, A. C. (2019). “A novel hybrid multi-objective metamodel-based evolutionary optimization algorithm”. In: Operations Research Perspectives 6, p. 100098. ISSN: 2214-7160. DOI: https://doi.org/10.1016/j.orp. 2019.100098.

Chen, C. (2022). Introduction to Structural Optimization: Truss Analysis Using the Principles of Finite Element Analysis, Genetic Algorithms and Python. Kindle Edition.

Deb, K., Gupta, H. (2006). “Introducing Robustness in Multi-Objective Optimization”. In: Evolutionary Computation 14.4, pp. 463–494. DOI: https://doi. org/10.1162/evco.2006.14.4.463.

Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley series in artificial intelligence. Addison-Wesley. ISBN: 9780201157673.

Jiang, P., Zhou, Q., Shao, X. (2020). Surrogate Model-Based Engineering Design and Optimization. Springer Singapore. ISBN: 978-981-15-0730-4. DOI: https://doi. org/10.1007/978-981-15-0731-1.

Park, Gyung-Jin *et al*. (2006). “Robust Design: An Overview”. In: AIAA Journal 44.1, pp. 181–191. DOI: https://doi.org/10.2514/1.13639.

Rasmussen, C. E., Williams, C.K.I. (2006). Gaussian Processes for Machine Learning. The MIT Press.

Soriano, H. L. (2005). Análise de estruturas: formulação matricial e implementação computacional. Editora Ciência Moderna. ISBN: 8573934522, 9788573934526

Taguchi, G. (1987). System of Experimental Design: Engineering Methods to Optimize Quality and Minimize Costs. Dearborn: American Suppliers Institute, pp. 653–655.

Tsutsui, S., Ghosh, A., Fujimoto, Y. (1996). “A robust solution searching scheme in genetic search”. Em: International Conference on Parallel Problem Solving from Nature. Springer, pp. 543–552.

Wackernagel, H. (2003). Multivariate Geostatistics: An Introduction with Applications. Springer Berlin Heidelberg. ISBN: 978-3-642-0791 1-5. DOI: https://doi.org/10.1007/ 978-3-662-05294-5.

Yang, Y. *et al*. (2022). "A Computationally Efficient Surrogate Model Based Robust Optimization for Permanent Magnet Synchronous Machines". In: IEEE Transactions on Energy Conversion 37.3, pp. 1520–1532.